SPECIAL PROJECT ON

ON

PSoC : Development Analysis & Options For Future Development

Prepared under the supervision of

Dr M.K. Deshmukh

(Electrical and Electronics Engineering Group)

By

Aalap Tripathy 2004P34PS208

For fulfillment of the requirements for

Electrical & Electronics Engineering Special Project (EEE GC 491)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE - PILANI, GOA CAMPUS

ZUARI NAGAR, GOA, INDIA

21st November, 2007

Abstract

This report presents the methodology used in preparation of a handbook and lab manual for Programmable System on Chip. This report must be read in conjunction with the PSoC Hand Book and Lab Manual which is attached separately. The experiments designed base on the compulsory discipline courses EEE GC 383-Communication Systems, ES GC 263 Microprocessor Programming, EEE GC 424 – Microelectronic Circuits, EEE GC 364 – Analog Electronics, EEE GC 371 – Electromechanical Energy Conversion, EEE GC 415 Digital Signal Processing. Also module descriptions as relevant in PSoC Designer and PSoC Express are included in this report.

Key Words

PSoC, PSoC Designer, PSoC Express, PSoC First Touch®, PSoC Evaluation Kits. SMP, MAC, Decimator, I2C Controller, Interrupt Controller, ADC, Amplifiers, Counters, DAC, Filters, PWM, Random Sequence Generators (PRS8,PRS16), Timers, I2C, SPI, UART, Del-Sigma-ADC,

Objective

This project aims to develop courseware for EEE GC 512 Embedded Systems and to incorporate Cypress Programmable System on Chip as the central element of the course. The author has been able to successfully develop and get validated a set of experiments which can be incorporated as modules in either a full semester course or as relevant modules at the end of regular CDC courses conducted at BITS, Pilani. A complete survey of available Application Notes have been done and categorized for relevancy to BITS Pilani courses.

Acknowledgement

No part of this work would have been possible without the active support and guidance of Dr M.K. Deshmukh. His commitment to ensuring relevancy of work and ease of use for future users made me choose this as topic for the special project.

I sincerely thank the encouragement by Mr M T Abhilash, Lab-in-charge, PSoC Lab, BITS Pilani Goa Campus who has allowed students like us access to lab facilities often at odd hours. I have possibly lost count for the number of times a talk with Mr Amalin Prince has let me out of desperation and made me continue work with renewed vigor.

Since, this project has been a culmination of my learning at BITS Pilani Goa Campus over four years, I would like to thank all my instructors especially Mrs Anita Agrawal, Mr Nitin Sharma, Mr A Khadke whose approach to problem solving has taught me many lessons some of which I have tried to incorporate in this work.

I would be erring if I did not thank my friends and the lab assistants especially Vijay Kumar Patil, Prakash Lamani, T K Prince who have helped me during the testing phase of the various experiments over the summer break.

Of course, none of this would ever have been possible without the support from Cypress Semiconductors, San Jose. Mr Ashish Garg, Strategic Marketing Engineer, Mr Kaushik Subhramaniam Narayanan have been our companions through this journey. Equally significant have been the support of Mr Kamal Gunsagar, Vice President, Business Development, Cypress Semiconductor, Mr Patrick Kane, Director, Cypress University Alliance, Mr Jeff Dahlin, Principal Application Engineer, Mr Dave van Ess, Principal Application Engineer and Chief of Technical Staff, Mr Ganesh Raja, the PSoC Master who have all regularly **reviewed the work** done and have made suggestions.

I sincerely hope that future readers of the accompanying lab manual and hand book make maximum use of the projects and be able to successfully design their own systems. At the end of this work, I have been convinced of one thing – "The possibilities are limitless. It is for us to go and explore".

Aalap Tripathy
aalap.tripathy@gmail.com
21st November, 2007

Table of Contents

Cover page *i*
Abstract *ii*
Keywords *ii*
Contents *iii*

1. Introduction 5
2. Contents of Work Done 7
   - 2.1 System Overview
   - 2.2 Basic Functionality
   - 2.3 Comparison with dsPIC
   - 2.4 Digital & Analog Functional Blocks
   - 2.5 SMP, MAC, Decimator
   - 2.6 I2C Controller, Interrupt Controller, Address Space
   - 2.7 Basic Module Description
   - 2.8 Advanced Module Description
   - 2.9 Specific Projects
     - 2.9.1 Blinking LEDs
     - 2.9.2 Controlling Blinking LEDs
     - 2.9.3 LCD Interfacing
     - 2.9.4 Digital Sine Wave Generation
     - 2.9.5 Manchester Code (generation)
     - 2.9.6 Single Pole IIR Filter
3. Survey of Application notes and Categorization 8
4. Recommendations for Students 22
5. Recommendation for Courseware Development 23
6. Direction for Future Work & Improvement 24
7. Sources for Information 25
8. Appendix – Sample of Handbook and Lab Manual 26

1. Introduction

PSoC (Programmable System-on-Chip) is a family of mixed-signal arrays first made by Cypress MicroSystems (CMS), a subsidiary of Cypress Semiconductors. This features a microcontroller and configurable integrated analog and digital peripherals. PSoC is a software configured, mixed-signal array with a built-in MCU core. The core is a Cypress proprietary, 8-bit Harvard architecture design called the M8C. PSoC has three separate memory spaces: paged SRAM for data, Flash memory for instructions and fixed data, and I/O Registers for controlling and accessing the configurable logic blocks and functions.

The PSoC contains an embedded microcontroller and is used in a wide variety of applications and market segments, including cell phones, portable media players, laptop computers, PDAs, white goods and industrial automation. Demand for the PSoC mixed-signal array has quadrupled over the past two quarters, driven in part by designs in handheld consumer devices. Cypress recently initiated production of PSoC devices in its high-volume Fab 4 facility in Bloomington, Minn., to keep up with increasing customer demand. PSoC is also manufactured at Cypress's Fab 2 plant in Round Rock, Texas, and will soon be produced at Grace Semiconductor Manufacturing Corp. as the result of a recent foundry agreement signed with Cypress.

As part of the Cypress Semiconductor's University Alliance Program, the students of BITS Pilani Goa Campus gained access to multiple evaluation boards, software tools which can be used to design systems. Though our technical training makes it possible for us to explore nuances of system specification and design, there appears to be lack of reading material and experiments specific to first time PSoC users which describes the system design process.

Further, though many of the individual modules viz. SMP, MAC, Decimator, I2C Controller, Interrupt Controller, ADC, Amplifiers, Counters, DAC, Filters, PWM, Random Sequence Generators (PRS8,PRS16), Timers, I2C, SPI, UART, Del-Sigma-ADCs etc have been covered in considerable detail in individual courses like EEE GC 383 –Communication Systems, ES GC 263 Microprocessor Programming, EEE GC 424 – Microelectronic Circuits, EEE GC 364 – Analog Electronics, EEE GC 371 – Electromechanical Energy Conversion, by students at BITS Pilani Goa Campus, no single point of reference exists which introduces students to the nuances of system design using these components in PSoC Designer Software.

Again, Cypress Semiconductor itself prefers users to ignore the component based specification of systems and define only inputs/outputs and the required transfer functions using PSoC Express while leaving the actual component selection, specification and use to the software. This aspect is explored in considerable detail using the newly acquired latest PSoC First Touch® Starter Kit.

So, this project uses existing application notes, example projects and freely available public resource material to prepare a primer suitable for a new user.

A **step by step approach** to designing of the lab experiments is presented with appropriate modifications which can be carried out as lab exercises. The content and subject of experiments has knowingly been chosen simple. Simple concepts used have been described in the theoretical analysis section. Suitable references to text books used as part of regular coursework has also been done to make a future user have a thorough understanding of the subject.

2. Contents of Work Done

1	Introduction	- Comparison with dsPIC - System Overview - Basic Functionalities - Digital & Analog Functional Blocks - SMP, MAC, Decimator - I2C Controller, Interrupt Controller, Address Space
2	Basic Module Description	1. ADC 2. Amplifiers 3. Counters 4. DAC 5. Filters 6. PWM 7. Random Sequence Generators (PRS8,PRS16) **8. Timers**
3	Advanced Module Description	1. I2C 2. SPI 3. UART 4. Del-Sigma-ADCs
4	Specific Projects	• Blinking LEDs • Controlling Blinking LEDs • LCD Interfacing • **Digital Sine Wave Generation** • **Manchester Code (generation)** • Single Pole IIR Filter

* Entries in **bold** have been included as samples in this project report

3. Survey of Application Notes and Categorization:

An attempt has been made here to enlist the relevant existing application notes written until August, 2007 and explore its relationship to BITS Course Work. The results of this study have been enumerated here for quick reference.

All of these applications can be used as Lab Oriented, Study Oriented, Computer Oriented Projects. In my opinion, since the **notes are complete material in themselves**, no attempt should be made at **plagiarizing** them, rather the subject of the projects should be to study the idea behind why certain modules were used and their suitability or unsuitability should be explored.

Mere reproduction of these notes will not have any practical impact for the learning process. Instead if these notes are used as **templates for further modification**, they will serve great practical significance. Of course, these notes can be given to first or second year students as single projects with or without academic credits.

Steps to using this Application Note Table :

1. The application notes must be read from bottom up. The earlier application notes as explored by the author are basic and are fundamental to our understanding of PSoC.
2. More attempts should be made to organize the early notes into a form of lab experiments and textual notes suitable for publishing.
3. As we move to the top of the list, the notes become more complex and assume a thorough understanding of the working of PSoC.
4. The author also contends that some intelligent students might be able to directly make sense of the top application notes. But, it is recommended that a study/project on the earlier application notes be done first.

Remarks :

This material is included with an expectation that future readers of this report will be able to have a ready reference in choosing projects, deciding whether projects they have in mind have in some way been already done. This note must be continually updated by project students every semester.

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2408	CapSense - Migrating from CSR to CSD		Ted Tsui	AN2041, AN2233a, AN2292
AN2407	USB and CapSense - PC Compatible USB CapSense Matrix Keyboard		Michael Macovetskyi, Ruslan Bachynskyy, Ryshtun Andrij	AN2233a, AN2292, AN2318, AN2352, AN2355
AN2405	Power - PSoC® IO Power Structure - Determining VOH and VOL at Partial Load		Dennis Seguine	
AN2401	Communication - Using the USBUART User Module	EEE GC 383 -Communication Systems	Svyatoslav Paliy, Vadym Grygorenko	
AN2399	Communication - Software Implementation of Universal Asynchronous Transmitter	EEE GC 383 -Communication Systems	Vadym Grygorenko, Volodymyr Sokil	
AN2397	CapSense - CapSense Data Viewing Tool		Vadym Grygorenko	AN2401, Using the USBUART User Module
AN2395	Thermistor Lookup Table Generation Tool		Petro Sasnyk	AN2017, AN2107, AN2260, AN2314
AN2404	General - PSoC® Implementation of a Newspaper Vending Machine Controller		Anant Aggarwal, Neha Joshi, Sachin Keswani, Shruti Richa	
AN2403	CapSense - Signal-to-Noise Ratio Requirement for CapSense Applications		Mark Lee	AN2233a, AN2277, AN2292, AN2318, AN2352, AN2355, AN2360, AN2394
AN2402	PSoC® Development Tools Selector Guide		N/A	
AN2398	CapSense - Waterproof Capacitance Sensing		Victor Kremin and Ruslan Bachunskiy	AN2352
AN2394	CapSense - CapSense Best Practices		Mark Lee	AN2233a, AN2277, AN2292,AN2318, AN2355, AN2360, AN2403
AN2393	CapSense - Migrating from CSR to CSA		Ted Tsui	AN2233a, AN2041

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2392	Multi-Context Switch Event Kernel		Uroš Platiše	
AN2389	Display and USB - Graphic OLED Display Demonstration Board With USB Interface	EEE GC 383 -Communication Systems	Michael Macovetskyi	AN2356
AN2388	USB - Voice Player with ADPCM Decoder	EEE GC 383 -Communication Systems	Ruslan Bachinskyy	
AN2385	Phase Controller with Current Limit	EEE GC 424 – Microelectronic Circuits	Kurt Labes	AN2025
AN2384	Using the MAC (multiply/accumulate) to compute scalar product of vectors	ES GC 263 Microprocessor Progr	Pengbo Sun, Alex Doboli, Eddie Currie	AN2032, AN2038
AN2383	Migrate from register oriented microprocessors to PSoC		Zoran Momirovic	
AN2380	Universal Wide-Range Signal Generator		Petro Kobluk	
AN2376	USB and Display - Four-Wire, Resistive-Type Touch Screen with USB Interface		Svyatoslav Paliy	AN2173
AN2375	General - Build a simple divider SYSCLK/4 using the digital features of the global digital interconnect (GDI) and row digital interconnect (RDI) on a PSoC® device		Wojciech Szyfelbein	
AN2374	Timers and Counters - Pulse counting with PSoC	EEE GC 383 -Communication Systems	J. Jayapandian	
AN2372	LED Testing and Control Using PSoC®	ES GC 263 Microprocessor Progr	David Johnson	
AN2369	Design Aids - Control an I2C Slave Device from PSoC Express		David Cooper	
AN2367	Analog - Differential Amplifier		Dave Van Ess	
AN2365	Design Aids - PSoC Express Timing App. Examples		Dave Funston	
AN2363	Design Aids - Sensor Calibration with PSoC Express™		Dave Funston	
AN2362	Capacitive Sensing - Wireless USB Remote Control			
AN2361	USB-Powered Battery Charger for NiCd/NiMH Batteries		Svyatoslav Paliy	AN2041, AN2107, AN2203, AN2260, AN2267, AN2267a
AN2360	Capacitive Sensing - Power Consumption and Sleep Considerations in Capacitive Sensing Applications		Mark Lee	
AN2359	Integrating an I2C Bootloader into PSoC Express		Dave Funston and M. Ganesh Raaja	AN2273, AN2273a
AN2358	Manchester Decoder Using PSoC®	EEE GC 383 -Communication	Philippe Larcher	AN2281, AN2325

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
		Systems		
AN2357	Power - Multi-Channel Fan Speed Control System		Volodymyr Sokil	AN2180, AN2246, AN2249, AN2314
AN2356	User Interface - Graphics Library for OSRAM's OLED Displays		Valeriy Kyrynyuk	AN2348
AN2355	Capacitance Sensing - Calibrating CapSense with the CSR User Module		Darrin Vallis	AN2233a, AN2277, AN2292, AN2318
AN2354	Power Management - Practical Application of the PSoC(R) Sleep Timer		Darrin Vallis	
AN2352	I2C-USB Bridge Usage		Valeriy Kyrynyuk	AN2304, CY3240-I2USB, CY3242-IOX
AN2351	Design Aids - Implementing Inter-Device Communications with PSoC Express™	EEE GC 383 -Communication Systems	David Cooper	AN2261
AN2349	Power Management - Increasing Output Power of a Switch Mode Pump		Vadym Grygorenko	AN2097, AN2180
AN2348	Tilt-Compensated Digital Magnetic Compass with Built-In Temperature Sensor and OLED Graphics Display		Vadym Grygorenko and Valeriy Kyrynyuk	AN2267, AN2272, AN2291, AN2314, AN2356
AN2347	Communication - PSoC(R)-Based Low-Cost, Intelligent Network: Sensor Applications	EEE GC 383 -Communication Systems	Andrew Smetana	AN2346
AN2346	Communication - PSoC(R)-Based Low-Cost, Intelligent Network: Physical and Data Link Layers	EEE GC 383 -Communication Systems	Andrew Smetana	AN2347, AN2086
AN2345	General - Simple Method to Generate Digital Signals with Variable Phase Shift Between		Victor Kremin, Ryshtun Andrij	
AN2344	Power - Multi-Cell Li-Ion/Li-Pol Battery Charger with Cell-Balancing and Fuel Gauge Function Support		Oleksandr Karpin	AN2180, AN2258, AN2294, AN2314
AN2343	Display - LCD Driver Based on the HT1621 Controller	EEE GC 491 – Special Projects	Andrew Smetana	AN2228
AN2342	Calculation - Building a Calculator with PSoC		Andrew Smetana	AN2343
AN2341	Algorithm - ArcTan as Fast as You Can		Dave Van Ess	
AN2340	Power Management - MAX1582 White LED Driver Emulation with PSoC®		Andrey Magarita	AN2041, AN2203, AN2316, AN2317, AN2331

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2339	Design Aids - Storing Calibration Factors into Flash Memory Within a PSoC Express(TM) Application		Dave Funston	AN2015
AN2338	Algorithm - Fast and Compact Unsigned Binary to BCD Conversion		Eugene Miyushkovich, Ryshtun Andrij	AN2112, AN2113
AN2336	Simplified FSK (Frequency Shift Keying) Detection	EEE GC 383 -Communication Systems	Dennis Seguine	
AN2335	Implement a PSoC (8-pin DIP) device suitable for sensing and controlling analog events from 0 to 5 volts		J. Jayapandian	
AN2334	Power Management - Multi-Cell Battery Voltage Measuring Device		Ruslan Bachinskyy	AN2041, AN2203
AN2333	Embedded State Machine Design for PSoC™ using 'C' Programming (Part III of III)	EEE GC 491 – Special Projects	Somsak Sukittanon	AN2329,AN2332
AN2332	Embedded State Machine Design for PSoC™ using 'C' Programming (Part II of III)	EEE GC 491 – Special Projects	Somsak Sukittanon	AN2329,AN2333
AN2330	USB-Powered Li-Based Battery Charger		Svyatoslav Paliy	AN2107, AN2267, AN2267a
AN2329	Embedded State Machine Design for PSoC™ using 'C' Programming (Part I of III)	EEE GC 491 – Special Projects	Somsak Sukittanon	AN2332, AN2333
AN2328	FIR Filtering with Application to Fast Hilbert Transform	EEE GC 415 – Digital Signal Processing	Somsak Sukittanon, Stephen Dame	
AN2326	Hardware Sequence Bitsteam Recognizer	EEE GC 383 -Communication Systems	Volodymyr Sokil	AN2249
AN2325	Serial Bit Receiver - HW Manchester Decoder	EEE GC 383 -Communication Systems	Volodymyr Sokil	AN2091,AN2249, AN2281,AN2236
AN2323	Build a PSoC(TM) Emulator into Your Board		Eddy Chu	
AN2321	Ground Isolation for ICE Debugger		JB Foreman	
AN2320	Offset Compensation for High Gain AC Amplifiers		Vadym Grygorenko	
AN2318	EMC Design Considerations for PSoC CapSense(TM) Applications		Mark Lee	
AN2317	MAX1698 White LED Driver Emulation with PSoC™	EEE GC 491 – Special Projects	Andrey Magarita	
AN2316	MAX1599 White LED Driver Emulation with PSoC(TM)		Andrey Magarita	
AN2315	3-Channel Filterbank in PSoC(TM)	EEE GC 415 – Digital Signal Processing	Somsak Sukittanon, Stephen Dame	
AN2314	Thermistor-Based Temperature Measurement in Battery Packs		Oleksandr Karpin	AN2017,AN2258,

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
				AN2260,AN2267, AN2294
AN2313	Pulse Oximeter		Serhiy Matviyenko	
AN2312	nth Order IIR Filtering Graphical Design Tool for PSoC(TM)	EEE GC 415 – Digital Signal Processing	Somsak Sukittanon, Stephen Dame	
AN2310	Comparator with Independently Programmable Hysteresis Thresholds		Dave Van Ess	
AN2309	Low-Cost, Two-Cell Li-Ion/Li-Pol Battery Charger with Cell-Balancing Support		Oleksandr Karpin	AN2107, AN2258, AN2267, AN2294
AN2308	Automotive - Remote Keyless Entry Car Alarm with Floating Code		Volodymyr Sokil	AN2268, AN2307
AN2307	Hardware Random Number Generator	EEE GC 383 -Communication Systems	Volodymyr Sokil	
AN2305	SPI-LIN Slave Bridge	EEE GC 383 -Communication Systems	Valeriy Kyrynyuk	
AN2304	I2C Port Expander with Flash Storage	EEE GC 383 -Communication Systems	Andrew Smetana	
AN2302	6-Channel DMX Dimmer		Petro Kobluk	
AN2301	Tachometer using a Switched Reluctance Rotation Sensor	EEE GC 491 – Special Projects	Victor Kremin	
AN2300	Non-Volatile Memory Controller with Real-time Clock		Svyatoslav Paliy	
AN2298	PSoC(TM)-Based USB Device Design By Example		John Hyde	
AN2294	Li-Ion/Li-Polymer Battery Charger with Fuel Gauge Function		Oleksandr Karpin	AN2108, AN2258, AN2267, AN2314
AN2292	Layout Guidelines for PSoC(TM) CapSense(TM)		Ryan Seguine, Mark Lee	
AN2291	Ultrasonic Vehicle Parking Assistant with LIN 2.0 Interface		Valeriy Kyrynyuk	
AN2287	Laser Power Meter, the PSoC(TM) Way		J. Jayapandian	
AN2286	Simulating a 555 Timer with PSoC(TM)	EEE GC 364 – Analog Electronics	Dave Van Ess	
AN2284	Sensing - Low-Cost EKG Pulsometer		Serhiy Matviyenko	AN2158
AN2283	Measuring Frequency	EEE GC 383 -Communication Systems	Dave Van Ess	
AN2282	Resonant Bridge Oscillators for Piezoelectric Buzzers		Andrey Magarita	
AN2281	Manchester Encoder Using PSoC(TM)	EEE GC 383 -Communication Systems	Ganesh Raaja	

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2279	Dynamic I2C Addressing Implemented with I2C Hardware User Modules	EEE GC 383 -Communication Systems	Chris Hogan	
AN2278	Automotive BLDC Motor Control for PSoC(TM)	EEE GC 371 - EMEC	unknown	
AN2277	Capacitive Front Panel Display Demonstration		Chris Hammer	AN2233A,AN2292
AN2276	Binary Weighted Single-Pole IIR Low-Pass Filters	EEE GC 415 – Digital Signal Processing	Dave Van Ess	AN2099
AN2274	Dynamic Reconfiguration Using 'C'	EEE GC 491 – Special Projects	Arnold Motley	
AN2273a	I2C Bootloader for PSoC(TM), 78-Byte Packet Transfer	EEE GC 383 -Communication Systems	Ernie Buterbaugh	AN2273
AN2273	I2C Bootloader for PSoC(TM), 16-Byte Packet Transfer	EEE GC 383 -Communication Systems	Ernie Buterbaugh	
AN2272	Magnetic Compass with Tilt Compensation		Vadym Grygorenko	
AN2269	Implement 9-Bit Protocol on the PSoC(TM) UART	EEE GC 383 -Communication Systems	Aubrey Kagan	
AN2268	Forward Error Correction using a Wireless USB Radio System-on-Chip (SoC) Modem	EEE GC 383 -Communication Systems	Andrew Smetana	
AN2267a	Single Cell Li-Ion Battery Charger using CY8C21xxx		Svyatoslav Paliy	AN2107
AN2267	Single Cell Li-Ion Battery Charger	EEE GC 491 – Special Projects	Svyatoslav Paliy	AN2107,AN2041
AN2266	16-Bit PWM/PWM-DACs using One Digital PSoC(TM) Block	EEE GC 383 -Communication Systems	Brian Miller	AN2199
AN2261	PSoC Express(TM) Primer: Introduction		Jon Pearson	
AN2260	Rapid NiCd/NiMH Battery Charger and DC Brushed Motor Controller for Autonomous Appliances	EEE GC 371 - EMEC	Victor Kremin	AN2041,AN2107, AN2168,AN2246
AN2258	Cell Balancing in a Multi-Cell Li-Ion/Li-Pol Battery Charger		Oleksandr Karpin	AN2107,AN2180
AN2257	Automotive Electromagnetic Compatibility (EMC) and PSoC	EEE GC 491 – Special Projects	William Parnis	
AN2256	Converting Projects from CY8C22x13 to CY8C24x23A		Jeff Dahlin	
AN2254	Low-Cost, RS-232 Level Translator	EEE GC 491 – Special Projects	Vitaliy Samusko	
AN2253	Intelligently “Bit-Bang” Debug Data		Michael Pail	
AN2252	LED Digit Displays - Large Quantities	EEE GC 364 – Analog Electronics	Luis Espinal	
AN2250	ECG Meter using PSoC(TM)	EEE GC 491 – Special Projects	Jens Altenburg	

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2249	PRS User Module as a One-Shot Pulse Width Discriminator and Debouncer	EEE GC 383 -Communication Systems	Ilya Mamontov	AN2108,AN2231
AN2247	Low CPU Consumption DTMF Detector	EEE GC 383 -Communication Systems	Rusian Bachinskyy	AN2122,AN2038, AN2027
AN2246	PWM Source - High Frequency, High Resolution	EEE GC 383 -Communication Systems	Victor Kremin	AN2041
AN2245	PWM Source - High Frequency, High Resolution	EEE GC 383 -Communication Systems	Victor Kremin	
AN2244	Smart Smoke Detector		Andrey Magarita	
AN2239	ADC Selection	EEE GC 364 – Analog Electronics	Dennis Seguine	AN2095,AN2219
AN2236	Converting Projects from CY8C24x23 to CY8C24x23A		Jeff Dahlin	
AN2233a	Capacitive Switch Scan	EEE GC 491 – Special Projects	Dennis Seguine	AN2277
AN2231	Ratemeter with a Precise Pulse Discriminator for Spectrometry		Ilya Mamontov	AN2144
AN2230	PreSoC: A Rational Preprocessor		Dave Van Ess	
AN2229	Multi-Functional Stepping Motor Driver	EEE GC 371 - EMEC	Victor Kremin	AN2161
AN2228	LCD Driving Methods using PSoC(TM)	EEE GC 364 – Analog Electronics	Svyatoslav Paliy	
AN2227	Brushless DC Motor Control	EEE GC 371 - EMEC	Andrey Magarita	AN2170
AN2225	Project Minimization for Version Control		John Lokanis	
AN2224	Lower Noise Continuous Time Signal Processing with PSoC		Dennis Seguine	AN2099,AN2216, AN226
AN2223	The Faux Op-Amp	EEE GC 491 – Special Projects	Dave Van Ess	
AN2222a	Flex-Pod Soldering Guide		Matt Basinger	
AN2221	Global Resources in PSoC Designer		Mohana Koteeswaren	
AN2219	Selecting PSoC Ground and Reference		Dennis Seguine	AN2017
AN2218	Large Memory Model Programming for PSoC		Khaled Boulos	
AN2216	Estimating PSoC Power Consumption		Onur Ozbek	
AN2214	Sonic Alarm	EEE GC 491 – Special Projects	Chris & Vincent Paiano	
AN2212	Dog Bark Eliminator (Bark -n- Squeal)		Chris & Vincent Paiano	
AN2209	Device Selection Guide for PSoC		Matt Basinger	
AN2208	Universal PID-Thermoregulator	EEE GC 364 – Analog Electronics	Andrew Smetana	AN2120,AN2148
AN2207	Lock-in Milliohmmeter	EEE GC 364 – Analog Electronics	Oleksandr Karpin	AN2028,AN2044,

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
				AN2120,AN2148, AN2158
AN2203	Programmable Analog High Current Source. PSoC Style	EEE GC 364 – Analog Electronics EEE GC 424 – Microelectronic Circuits	Dave Van Ess	AN2089
AN2200	Stud Finder		Chris & Vincent Paiano	
AN2199	DAC With Analog Modulator	EEE GC 364 – Analog Electronics EEE GC 383 -Communication Systems	Ganesh Raaja	AN2117
AN2197	Stepper Motor Driver for Smart Gauges	EEE GC 371 - EMEC	Victor Kremin	AN2161
AN2192	Digital Bipolar Power Chopper	EEE GC 491 – Special Projects	Chris & Vincent Paiano	
AN2187	Audible Clock	EEE GC 364 – Analog Electronics	Chris and Vincent Paiano	
AN2186	Acoustic Glass Break Detector	EEE GC 491 – Special Projects	Vadym Grygorenko	
AN2182	Radio Race Control System Encoder	EEE GC 383 -Communication Systems	Chris and Vincent Paiano	
AN2180	Switch Mode Pump in a Step-Down Converter Using PSoC		Andrey Magarita	
AN2178	Yet Another PSoC-Based Oscilloscope	EEE GC 364 – Analog Electronics	Andrea Giacosi	
AN2177	Ultra-WideBand RADAR Test Platform		Peter A. Stephens	
AN2173	Touch Screen Control and Calibration - Four-Wire, Resistive		Svyatoslav Paliy	
AN2170	3-Phase Brushless Direct Current Motor Driver with Hall-Effect Sensor	EEE GC 371 - EMEC	Andrey Magarita	
AN2168	Understanding Switched Capacitor Filters	EEE GC 364 – Analog Electronics	Dave Van Ess	
AN2166	1-Wire User Modules (Introduction)		Wes Randall	
AN2165	Implementing Direct Sequence Spread Spectrum in the PSoC	EEE GC 383 -Communication Systems	Kristopher Young	
AN2163	Temperature Measurement with a 1-Wire Digital Sensor	EEE GC 364 – Analog Electronics	Onur Ozbek	
AN2162	Using the PSoC Invention Board		Andrew Page	
AN2161	Voltage-to-Frequency Converter	EEE GC 364 – Analog Electronics	Victor Kremin	AN2041,AN2044
AN2159	Analog Multiplication with PSoC	EEE GC 491 – Special Projects	Victor Kremin	AN2041,AN2044, AN2161

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2158	Optical PulsOmeter with PSoC	EEE GC 491 – Special Projects	Victor Kremin	AN2041,AN2042, AN2058,AN2152
AN2157	Three-Phase Sine Wave Generator	EEE GC 364 – Analog Electronics	Uros Platise	AN2141
AN2156	A Switched Capacitor Comparator with Programmable Hysteresis		Dave Van Ess	AN2044,AN2041, AN2108
AN2155	EMI Design Considerations for PSoC		Dennis Seguine	
AN2154	Voltage Monitoring and Sequencing with PSoC	EEE GC 364 – Analog Electronics	Ernie Buterbaugh	
AN2153	Model Rocketry: Air-Starting Rocket Motors	EEE GC 371 - EMEC	Joe Peck	
AN2152	Graphics LCD and PSoC Interface	ES GC 263 Microprocessor Progr	Svyatoslav Paliy	
AN2148	Measuring Temperature Using a Thermocouple	EEE GC 364 – Analog Electronics	Ganesh Raaja	AN2099,AN2038, AN2101
AN2147	Interfacing to a Graphics LCD from PSoC	EEE GC 364 – Analog Electronics	Pham Minh Tri	
AN2146	Model Rocketry In-Flight Digital Imaging	EEE GC 491 – Special Projects	Joe Peck	
AN2145	Implementing Hardware Quadrature Phase Decoders	EEE GC 383 -Communication Systems	Edwin Olson	
AN2144	Window Discriminator	EEE GC 383 -Communication Systems	vikram_cr@hotmail.com	AN2108
AN2141	Glitch-Free PWM	EEE GC 383 -Communication Systems	uros.platise@ijs.si	
AN2138	3-Wire Interface for LCD Display	EEE GC 364 – Analog Electronics	graaja@eth.net	
AN2137	Subscriber Pulse-Metering Detector	EEE GC 383 -Communication Systems	djzoheir@caramail.com	
AN2136	24, 8-Bit Hardware PWMs in a Single PSoC	EEE GC 383 -Communication Systems	GMaudens@SASCO.de	
AN2135	Calling Functions Using a Vector Table in C		CyPro@opctechnology.com	
AN2134	PSoC Programmer for CY8C26XXX Devices		alex.mol@tiscali.it	AN2014, AN2026
AN2133	Autonomous Robot	EEE GC 491 – Special Projects	soccer_fan1@yahoo.com	AN2086
AN2132	Multithreading on the PSoC	ES GC 263 Microprocessor Progr	Justin@holmesweb.net	
AN2131	Migrating Projects to CY8C27xxx in PSoC Designer 4.0		Andrew Best	
AN2130	Dynamic Re-configuration Using Trace	EEE GC 491 – Special Projects	Vincent Aubineau	

THE CYPRESS UNIVERSITY ALLIANCE

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2129	Interfacing Assembly and C Source Files	ES GC 263 Microprocessor Progr	Jerel Byrd	
AN2128	3-Wire Interface for a 4-Digit LED Display	EEE GC 383 -Communication Systems	graaja@eth.net	
AN2125T	Blackjack Game (Turkish Version) - Preliminary		Cihan Fidan	
AN2125	Standard - Blackjack Game (English Version) - Preliminary	EEE GC 491 – Special Projects	Cihan Fidan	
AN2124	Morse Decoder for the PSoC	EEE GC 383 -Communication Systems	Melchor A. Varela Morales	
AN2122	Standard - DTMF Detector	EEE GC 383 -Communication Systems	Victor Kremin	
AN2121	Using PSoC Internal Resistors for I2C Communications - Preliminary	EEE GC 383 -Communication Systems	Jason A. Goldstein	
AN2120	RTD Temperature Measurement		M. Ganesh Raaja	
AN2118	Telephone Call Logger - Preliminary		M.Zeki SONMEZ	
AN2117	DAC-11 - Preliminary	EEE GC 364 – Analog Electronics	M. Ganesh Raaja	
AN2116	PC to PSoC Communications with Scrolling LCD Message	EEE GC 383 -Communication Systems	Onur OZBEK	
AN2115	Generate Triangle and Trapezoid Waveforms with a Switched Capacitor -Preliminary		Sigurd Peterson	
AN2114T	Playing Musical Notes Using Buzzer (Turkish Version)		Cihan Fidan	
AN2114	Playing Musical Notes Using Buzzer (English Version)	EEE GC 364 – Analog Electronics	Cihan Fidan	
AN2113P	Math Programs (Portuguese Version)		Harald W. Cintra	
AN2113	Math Programs (English Version)	ES GC 263 Microprocessor Progr.	Harald W. Cintra	
AN2112	Binary To BCD Conversion - Preliminary	ES GC 263 Microprocessor Progr.	Ganesh Raaja	
AN2111	Heterodyne with Quadrature Outputs, PSoC Style	EEE GC 383 -Communication Systems	Dave Van Ess	
AN2110	Use an ADCINC12 and Get a Free PWM8, While Supplies Last!	EEE GC 364 – Analog Electronics	Dave Van Ess	
AN2109	The Direct Digital Synthesis Generator	EEE GC 491 – Special Projects	Victor Kremin	
AN2108	Hysteresis Comparator with PSoC	EEE GC 364 – Analog Electronics	Mohana Koteeswaren	
AN2107	A Multi-Chemistry Battery Charger	EEE GC 491 – Special Projects	Victor Kremin	
AN2106	Simple PC Oscilloscope (Using TX8 and SAR6)	EEE GC 364 – Analog Electronics	Mehmet Z Sonmez	
AN2105	Pyroelectric Infrared Motion Detector, PSoC Style		Dave Van Ess	

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2104	Dynamic Re-configuration: Getting Started - Preliminary		Frank Berkner	
AN2103	Measuring an Input PWM	EEE GC 383 -Communication Systems	Steve Gerber	
AN2102	Protective Controller for a Refrigerator (English Version)		Azim Gadzhiev	
AN2101	Unsigned Division Routines	ES GC 263 Microprocessor Progr.	Ganesh Raaja	
AN2100	Bootloader: PSoC	ES GC 263 Microprocessor Progr.	Andrew Smetana	
AN2099a	Single-Pole IIR Filters. To Infinity And Beyond! (Japanese Version)	EEE GC 415 – Digital Signal Processing	Dave Van Ess	
AN2099	Single-Pole IIR Filters. To Infinity And Beyond!	EEE GC 415 – Digital Signal Processing	Dave Van Ess	
AN2098	FSK Generator using the PSoC Device	EEE GC 383 -Communication Systems	Andrew Page	
AN2097	Switch Mode Pump	EEE GC 364 – Analog Electronics	Mohana Koteeswaren	
AN2096	Using the ADCINC12, It's as Easy as A D C		Dave Van Ess	
AN2095	Logarithmic Signal Companding.	EEE GC 364 – Analog Electronics	Dave Van Ess	
AN2094	PSoC I/O Pin-Port Configuration	ES GC 263 Microprocessor Progr.	Mehmet Z Sonmez	
AN2093	Keypad Scan using ADC (SAR6)	ES GC 263 Microprocessor Progr.	Mehmet Z Sonmez	
AN2092	Infrared Learner (Remote Control)	ES GC 263 Microprocessor Progr.	Mehmet Z Sonmez	
AN2091	RC5 Codec	EEE GC 383 -Communication Systems EEE GC 415 – Digital Signal Processing	Victor Kremin	
AN2090	VECTOR'SoC: A 1 GHz Vectorial Network Analyzer	EEE GC 383 -Communication Systems	Robert Lacoste	
AN2089	Programmable Bipolar Analog Current Source. PSoC Style	EEE GC 424 – Microelectronic Circuits	Dave Van Ess	
AN2088	Programmable I2C Addressing	EEE GC 383 -Communication Systems EEE GC 391 Dig Elec. ES GC 263 Microprocessor Progr.	Arnold Motley	

THE CYPRESS UNIVERSITY ALLIANCE

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2087	Motor Tachometer Speed Calculation Using Hardware Timer Capture Feature	EEE GC 371 - EMEC	Arnold Motley	
AN2086	Digitally Controlled Sine and Square Wave Generation	EEE GC 491 – Special Projects	Jerry Wasinger	
AN2047	Ultrasound Motion Sensor	EEE GC 491 – Special Projects	Victor Kremin	
AN2046	Real-Time Operation System for PSoC MCUs		Edward Nova	
AN2045j	Designing a Compact and Flexible LIN Controller (Japanese)		Philippe Larcher	
AN2045	Designing a Compact and Flexible LIN Controller		Philippe Larcher	
AN2044	Signal Rectification, using Switched Capacitor Modulators	EEE GC 364 – Analog Electronics	Dave Van Ess	
AN2043	Real-Time Clock in PSoC	EEE GC 491 – Special Projects	Corey Wilner	
AN2042	Multifunctional Optical Sensor	EEE GC 364 – Analog Electronics	Victor Kremin	
AN2041	Understanding Switched Capacitor Analog Blocks	EEE GC 491 – Special Projects	Dave Van Ess	
AN2040	Entering and Leaving 24 MHz Operation for CY8C25xxx/26xxx		Cy Apps	
AN2039	Advanced Power Management for CY8C25xxx/26xxx		Cy Apps	
AN2038	Signed Multi-Byte Multiplication	EEE GC 391 Dig Elec. ES GC 263 Microprocessor Progr.	Dave Van Ess	
AN2037	8 PDIP Produces 100 kHz Pseudo Random White Noise (with a Six-Hour Period)	EEE GC 391 Dig Elec. ES GC 263 Microprocessor Progr.	Jerry Wasinger	
AN2036	A Circular FIFO, PSoC Style	EEE GC 491 – Special Projects	Dave Van Ess	
AN2034	Keypad Scan, PSoC Style	EEE GC 391 Dig Elec. ES GC 263 Microprocessor Progr.	Dave Van Ess	
AN2033	Data Port Bit Manipulation with the PSoC MCU	EEE GC 391 Digital Electronics & Computer Organization	Darrin Vallis	
AN2032	Unsigned Multiplication	EEE GC 391 Dig Elec. ES GC 263 Microprocessor Progr.	Dave Van Ess	
AN2031	Adjustable Sallen and Key Low-Pass Filters	EEE GC 364 – Analog Electronics	Dennis Seguine	
AN2030	Adjustable Sallen and Key High-Pass Filters	EEE GC 364 – Analog Electronics	Dennis Seguine	
AN2028	Ohmmeter	EEE GC 364 – Analog Electronics	Dave Van Ess	
AN2027	Using the PSoC Microcontroller External Crystal Oscillator	EEE GC 491 – Special Projects	Jeff Dahlin	
AN2026b	In-System Serial Programming Protocol CY8C24794 and CY8C29xxx	EEE GC 391 Dig Elec. ES GC 263 Microprocessor Progr.	Jeffrey Stewart	
AN2026a	In-System Serial Programming Protocol CY8C21/22/24/27		Jeffrey Stewart	

Cypress PSoC Application Note Description & Relevancy to BITS Course Work

Application Note Number	Description	Relevance to Course	Author	Associated Application Notes
AN2026	In-System Serial Programming (ISSP) Protocol		Jeffrey Stewart	
AN2026	In-System Serial Programming (ISSP) Protocol (Japanese)		Jeffrey Stewart	
AN2025	CTCSS Carrier Generation with a PSoC	EEE GC 383 -Communication Systems	Jeff Dahlin	
AN2024	Polyphonic Piano	EEE GC 491 – Special Projects	Dave Van Ess	
AN2021	What is an Invalid Memory Reference	ES GC 263 Microprocessor Progr.	Craig Nemecek	
AN2020	Redundant Fan System	EEE GC 491 – Special Projects	Mark Francis	
AN2018	Care and Feeding of ICE Pods		Craig Nemecek	
AN2017	A Thermistor-Based Thermometer, PSoC Style		Dave Van Ess	
AN2016	The Cypress MicroSystems Device YProgrammer		Craig Nemecek	
AN2015	Flash APIs	ES GC 263 Microprocessor Progr.	Warren Snyder/Jon Perrin	
AN2014	Design for In-System Serial Programming (ISSP)		Mark Hastings	
AN2013	UART Receiver Errata Workaround	EEE GC 383 -Communication Systems	Jon Perrin/Steve Roe	
AN2012	Adjusting PSoC Trims for 3.3 Volt Operation with PSoC Designer Versions 2.xx		Jeff Dahlin	
AN2011	PSoC Pup Example Projects	EEE GC 491 – Special Projects	Cy Apps	
AN2010	Getting Started with PSoC (READ THIS FIRST)		Jeff Dahlin	

4. Recommendation for Students:

1. Review CYU Course Material Module 1
2. Review CYU Course Material Module 2
3. Review CYU Course Material Module 3
4. Review CYU Course Material Module 4
5. Visit http://www.easypsoc.com/book
6. Read PSoC 101 – AN 2010 – Getting started with PSoC, Jeff Dahlin
7. Visit http://www.psocdeveloper.com and perform example projects
8. Review CYU Course Material Module 1 Again
9. Perform Blinking LED's example from manual and associated problems
10. Perform LCD Interfacing from manual and associated problems
11. Perform Sine Wave Generation from manual and associated problems
12. Perform Manchester Code Example from manual and associated problems

5. Recommendation for Courseware Development :

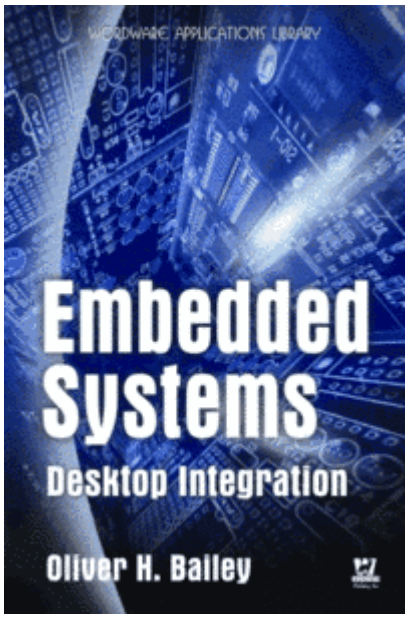
The following are the recommendations of the author and features available in the current version of the lab manual.

1. Inclusion of Timer, Counter examples as compulsory components in Digital Electronics and Computer Organization Course
2. Inclusion of Assembly Coding Examples in Microprocessor Programming and Interfacing Course
3. Inclusion of C Coding Exercises for specific applications as lab oriented courses specific to PSoC
4. Specific modules as selected by students/instructor performed as COP/SOP/LOP/Special Project which involves the following :
   a. Developing simplified descriptions of module parameters. Mere reproduction of datasheet entries will not be useful
   b. Quoting specific application notes where the modules have been used with explanation of how relevant they have been, performance testing and implementation of those application notes with comments on improvement of parameter values.
5. Selection of application notes and categorization as per difficulty level. One section of the lab manual experiment must include screenshots of implementation in simple steps. The associated instructions must be available nearby.
6. Problems similar to the subject of the experiment must be chosen, solved and tested before inclusion in lab manual.
7. A non-technical explanation of the subject of the experiment must be included in theoretical analysis of the problem.
8. Suitable references to text books in use must be recommended.
9. Recommendations be made for purchase/procurement of additional items from Cypress through University Alliance by the supervisor.

10. Improvement & Future Work:

1. This hand book covers only the basic design elements and modules in the PSoC Designer Software. Modifications must be done using PSoC Express and many more sections keeping to the overall framework must be added.
2. Description of PSoC Internal Configuration as described in CYU Course Material may be included.
3. More experiments may be designed keeping the format in view – description and screenshot as in software.
4. Suitable problems can be designed for each experiment.
5. This manual needs to continually updated to keep in tune with the work being done by different PSoC Users. Assistance from developers at http://www.psocdeveloper.com must be taken.

7. Sources for Information/References



mikro easyPSoC

http://www.easypsoc.com/book/

Step By Step Understanding Material

1. CY Technical Reference Manual
2. CY Example Projects
3. CY Courseware
4. Embdedded System Desgin, Oliver Bailey
5. Embedded Systems – Desktop Integration, Oliver Bailey
6. http://www.psocdeveloper.com
7. http://www.time-lines.com/
8. http://easypsoc.com/book/
9. http://www.circuitcellar.com/library/print/0804/Eady169/index.htm
10. There is a wealth of PSoC user module information contained within the PSoC Designer IDE. All of the user module datasheets, are just a click away, they include everything needed to know to deploy the module and a sample code snippet that can be cut and paste into a PSoC project.
11. CY App Team Notes

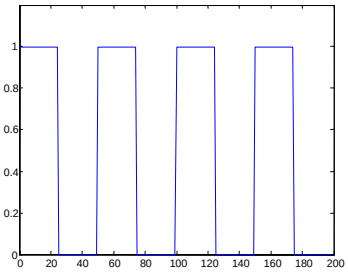
8. Appendix :

The following sections derived from the handbook are attached for reference of the reader. Please acquire a copy of the manual under development for a complete picture.

1. Timer Module Description
2. Digital Sine Wave Generation
3. Manchester Code Generation

EXPERIMENTS – LEVEL 2
SIGNAL GENERATION
Generate a fixed frequency Sine Wave

Theoretical Analysis (AN 2086)



The fourier series of a square wave is given by :

$$w(t)= a_0 + \sum_{n=1}^{\infty} a_n \cos nw_0t + b_n \sin nw_0t$$

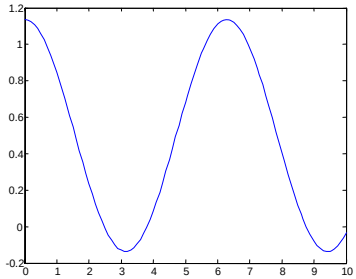
$$\text{i.e } a_0=\frac{1}{T_0}\int_{T_0} w(t)dt = \frac{1}{T_0}\int_{-T_0/4}^{+T_0/4} dt = 1/2$$

$$a_n=\frac{2}{T_0}\int_{-T_0/4}^{T_0/4}\cos n\omega_0 t dt = \frac{2}{n\Pi}\sin\left(\frac{n\Pi}{2}\right)$$

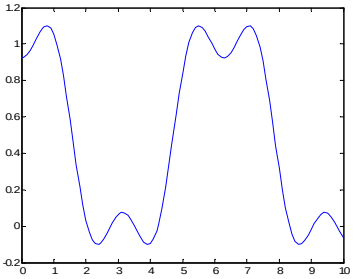
$$b_n=\frac{2}{T_0}\int_{-T_0/4}^{T_0/4}\sin n\omega_0 t dt = 0$$

$$\text{So, } w(t) = \frac{1}{2}+\frac{2}{\pi}\left(\cos\omega_o t-\frac{1}{3}\cos 3\omega_0 t+\frac{1}{5}\cos 5\omega_0 t-\frac{1}{7}\cos 7\omega_0 t+\frac{1}{9}\cos 9\omega_0 t-..+\right)$$

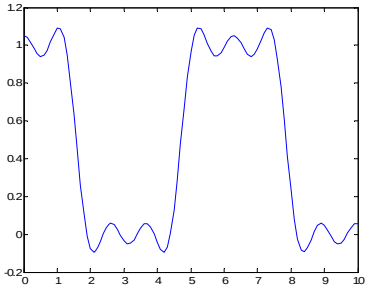
Assuming $\omega_0=1$



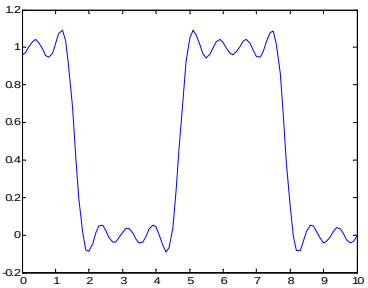
```
t = 0:.1:10;
y = 1/2+(2/pi)*(cos(t));
plot(t,y);
```



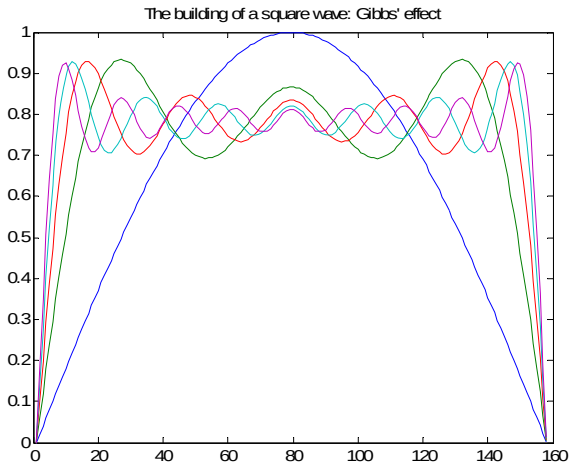
```
t = 0:.1:10;
y = 1/2+(2/pi)*(cos(t) -(1/3)*cos(3*t));
plot(t,y);
```



```
t = 0:.1:10;
y = 1/2+(2/pi)*(cos(t) -
(1/3)*cos(3*t)+(1/5)*cos(5*t));
plot(t,y);
```



```
t = 0:.1:10;
y = 1/2+(2/pi)*(cos(t) -
(1/3)*cos(3*t)+(1/5)*cos(5*t));
plot(t,y);
```




```
MATLAB Code (For Verification)
t = 0:.02:3.14;
y = zeros(10,length(t));
x = zeros(size(t));
for k=1:2:19
    x = x + sin(k*t)/k;
    y((k+1)/2,:) = x;
end
plot(y(1:2:9,:)')
title('The building of a square wave: Gibbs'' effect')
```

Assuming w(t) = $\frac{1}{2}+\frac{2}{\pi}\left(\cos\omega_o t-\frac{1}{3}\cos 3\omega_0 t+\frac{1}{5}\cos 5\omega_0 t-\frac{1}{7}\cos 7\omega_0 t+\frac{1}{9}\cos 9\omega_0 t-..+\right)$

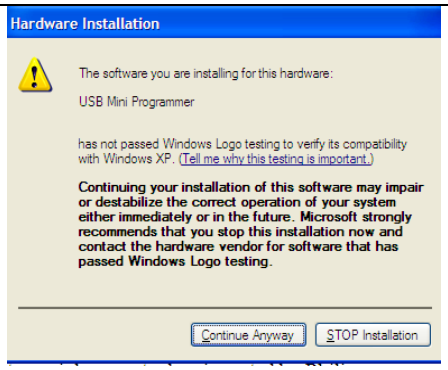
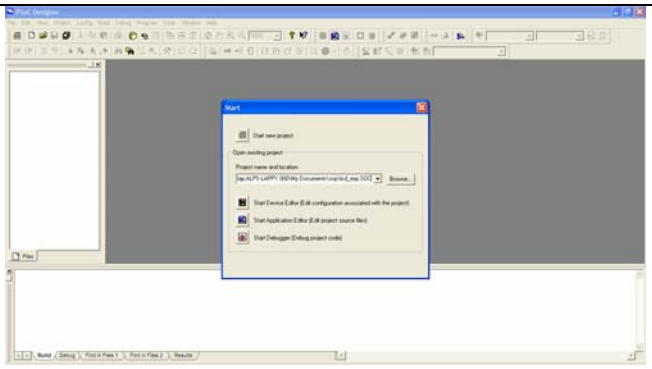
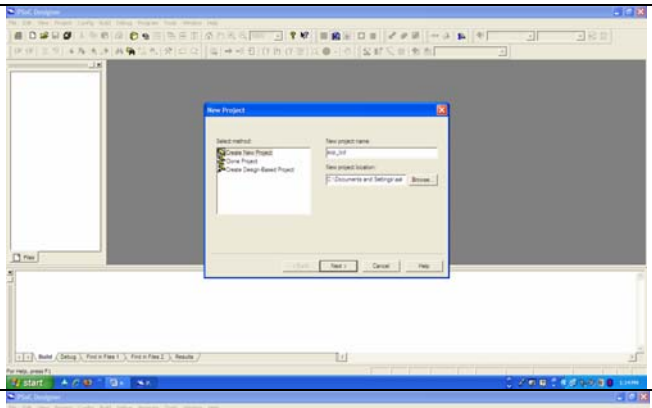
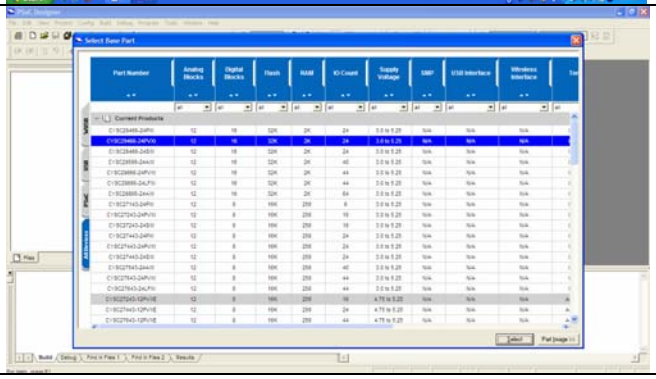
And $\omega_0=2\pi f$, Let us assume f=1 unit = 1 Khz (say)

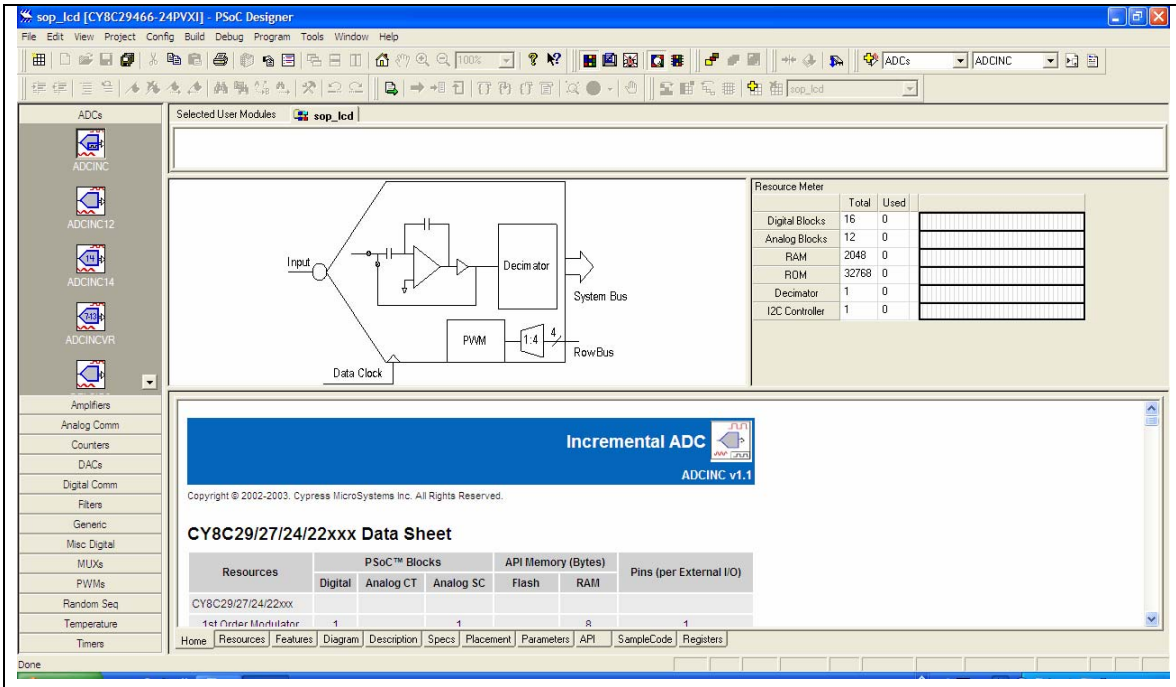
So, w(t) = $\frac{1}{2}+\frac{2}{\pi}\left(\cos 2\pi t-\frac{1}{3}\cos 3\pi t+\frac{1}{5}\cos 5\pi t-\frac{1}{7}\cos 7\pi t+\frac{1}{9}\cos 9\pi t-..+\right)$

To generate a sine wave from a given square wave, we need to pass this through a Band Pass Filter

The following simplification (based on AN2086) has the following features:

1. Use the BPF2 User module datasheet to determine the filter parameters such that:
   - Center frequency = 1Khz
   - Q=4
   - Oversampling Rate = 50
2. Two BPF2 filters are used to obtain accuracy
3. An 8 bit counter used to obtain a square wave of 1Khz frequency
4. For demonstration purpose, we are also using a 16 bit counter (fed at 24 Mhz) to implement a divide by 200. *This generates the clock input for the programmable gain amplifier*
5. Output of Counter8_1 fed to pin P0[0]
6. This is externally connected (Explore advantages and disadvantages of internal connection if possible) to the input of a programmable gain amplifier (PGA) in the analog module section
7. To avoid saturation of the output sine wave, gain of PGA set to 0.75 (Examine practical limits of PGA gain when the final output becomes unidentifiable). *Note that saturation of the square wave is meaningless because after clipping this would still be square.*

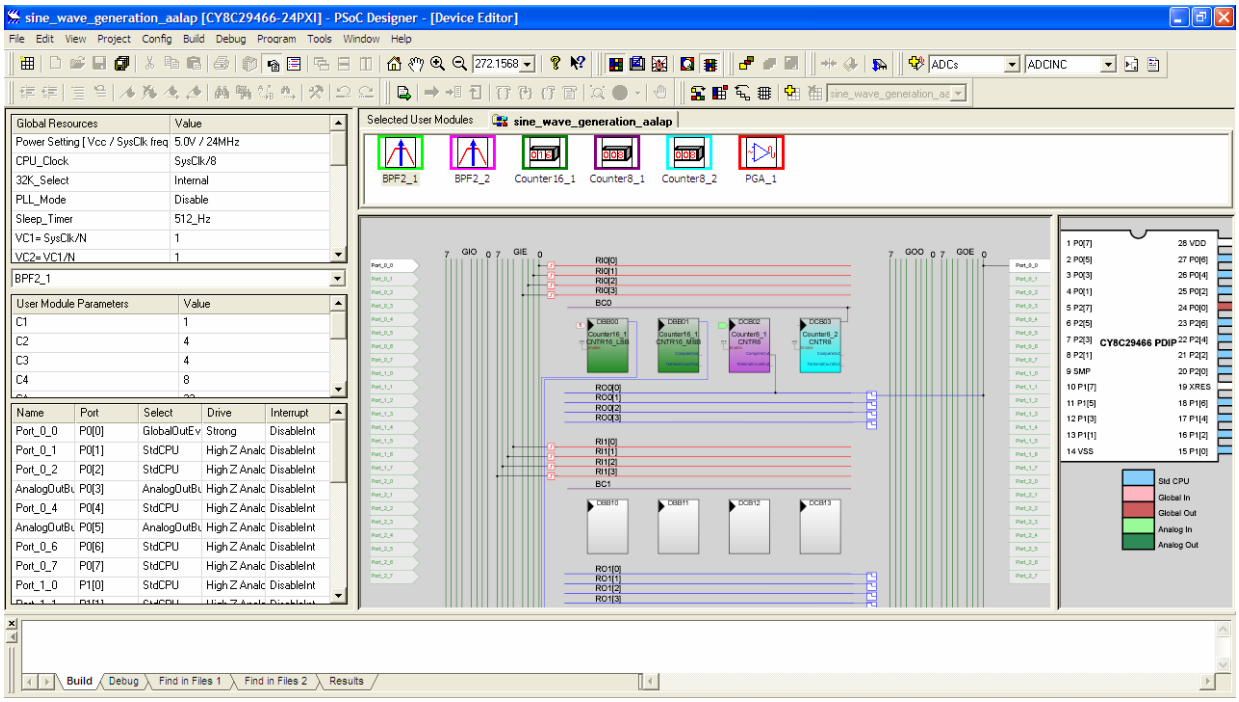
8. The Sine wave output is finally obtained at P0[5]	
1. Plug in the USB Connector to the PSoC Mini-Programming Kit. For the first time a new driver installation will take place. 2. We place CY8C29466-24PXI in the dock Programming the PSoC is a 2 step process • Develop the Code in PSoC Designer • Download code to the device	
3. Cypress Microsystems | PSoC Designer 4. Choose New Project	
5. Type Project Name	
6. Incase correct part is not chosen, use View Catalog 7. Generate Main File using C 8. By default Assembler option is selected. In case the C Compiler option is disabled, please goto Tool → Options → Compiler and enter the ImageCraft Serial Number available in the Lab 9. The Characters after the hyphen indicate the part of packaging.	
10	



Notes :

1. Left pane shows preconfigured elements. They can be selected by highlighting the component → Right Click → Select
2. The resource meter on upper right side shows what part resources are used and available

11.

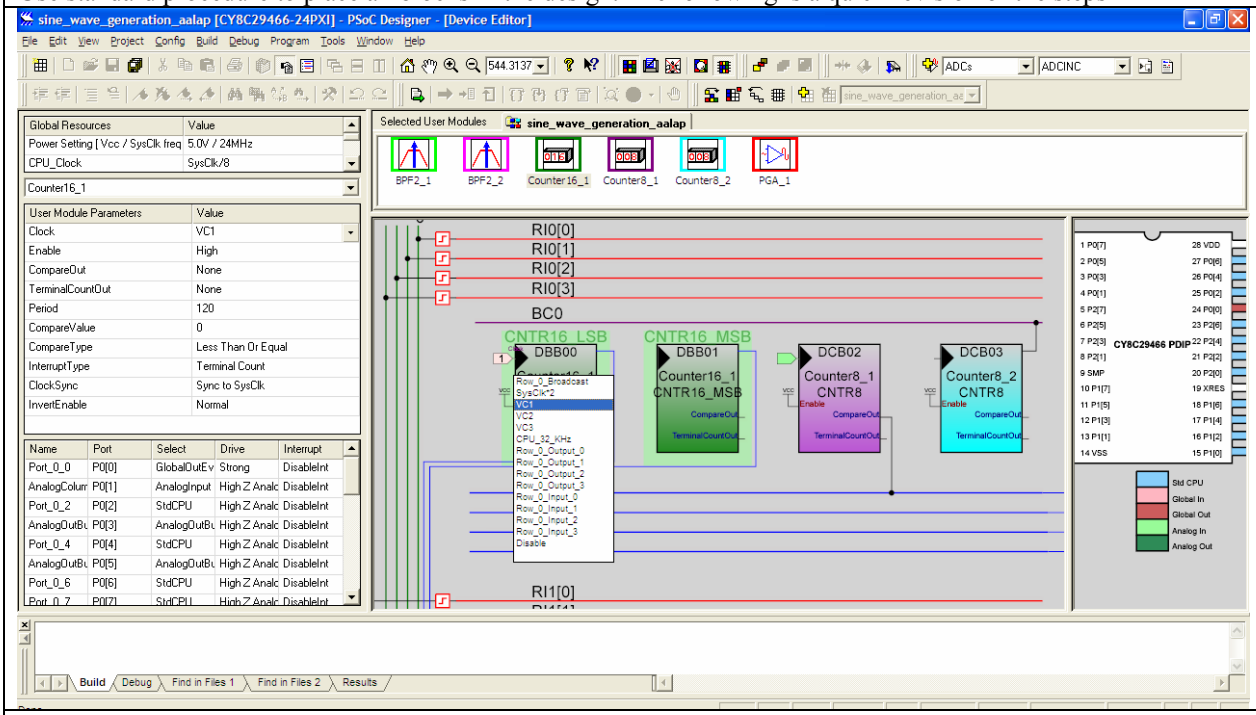


The following modules are placed :

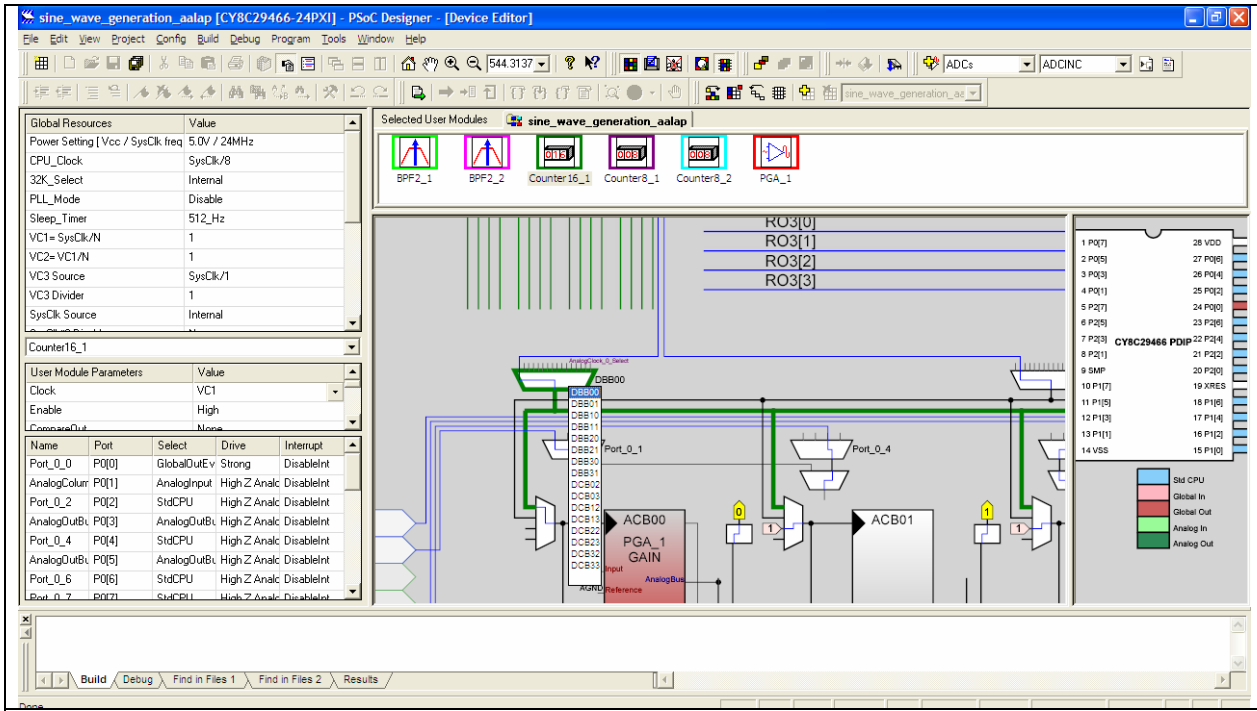
- Filters → BPF2_1 & BPF2_2
- Counters → Counter16_1
- Counters → Counter8_1 & Counter8_2
- Amplifiers → PGA_1

12. Switch from the **user module view** to the **Interconnect view**

Use standard procedure to place all blocks in the design. The following is a quick revision of the steps

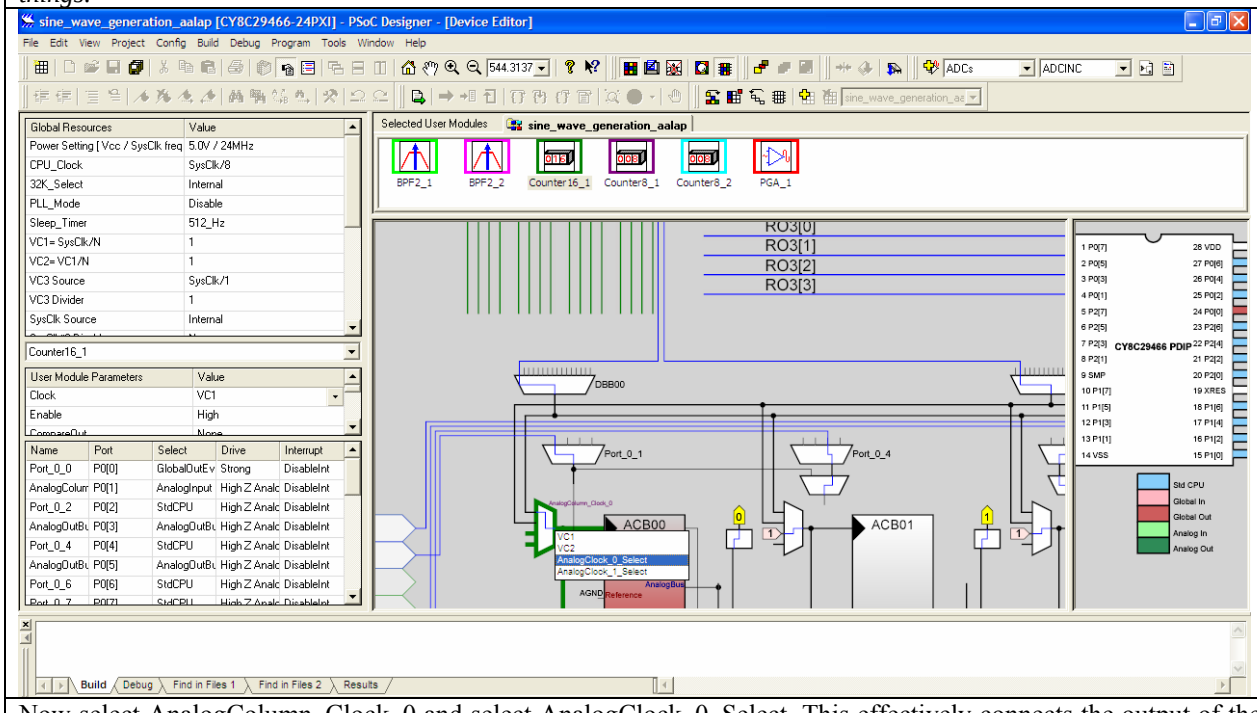


13. Click on the clock input of the 16 bit counter. Select VC1 as shown. This makes it get a clock signal of SysClk=24 Mhz. *You can use VC3 to do this if you only need 8 bit divider. This will keep the digital block free for other things.*



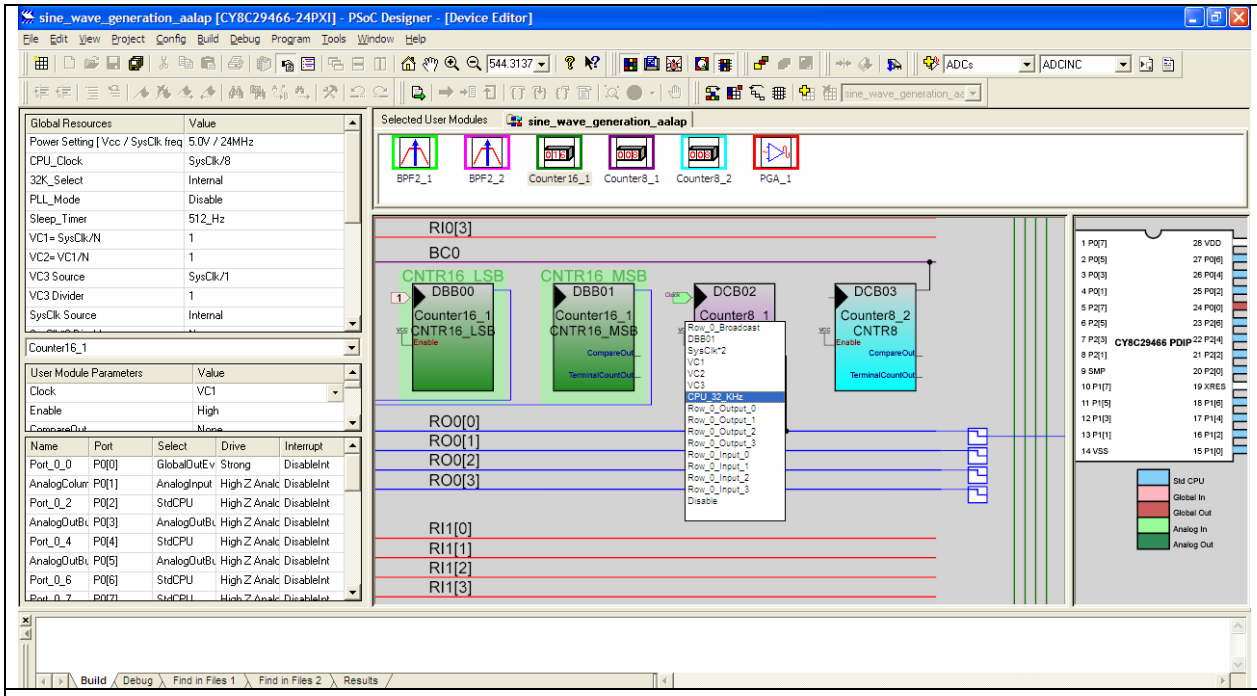
14. We will give the LSB of the 16 bit counter (divide by 8) as clock to the PGA. Scroll down to AnalogClock_0_Select and select DBB00 (where the LSB section of the 16 bit counter is placed).

Note : If dividing by less than 256, then we can use an 8-bit counter. By using an 8-bit you will save digital blocks for other things.



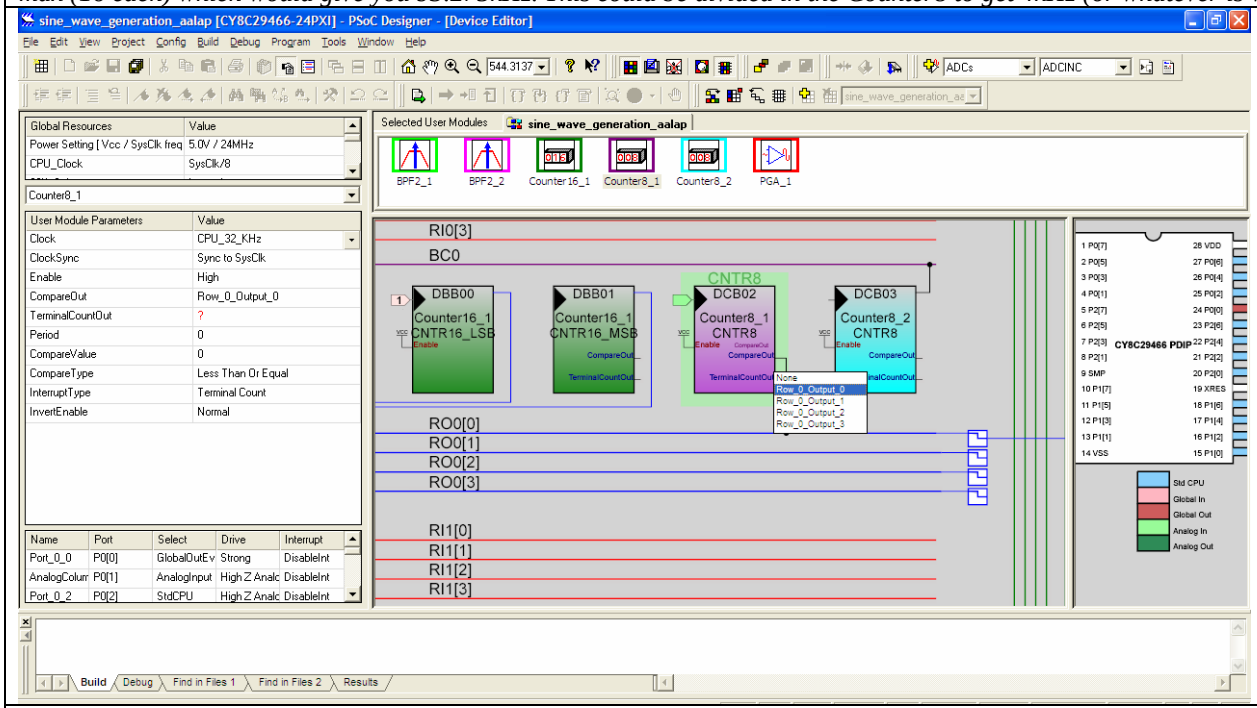
Now select AnalogColumn_Clock_0 and select AnalogClock_0_Select. This effectively connects the output of the LSB of the 16 bit counter to the clock input of the Programmable Gain Amplifier.

Once the default experiment is over, one could try connecting the MSB of the 16 bit counter (here connected to AnalogColumn_Clock_1) and give it as clock input to the Programmable Gain Amplifier. Note the change in the outputs.



Scroll up again to the first 8 Bit Counter (Counter8_1) This is the counter which actually is being used here to generate the square wave of 4 Khz frequency (32 Khz/4) Select CPU_32_Khz as the clock input. Other options might be tried once the basic experiment is over.

Note : The 32 kHz clock is not very accurate, so maybe it is not a good choice. The accuracy of the CPU_32kHz clock is from 15kHz to 64kHz. So, the 4 kHz could be anywhere from 2kHz to 8 kHz. I suggest dividing down VC1 and VC2 by the max (16 each) which would give you 93.275kHz. This could be divided in the Counter8 to get 4kHz (or whatever is wanted).

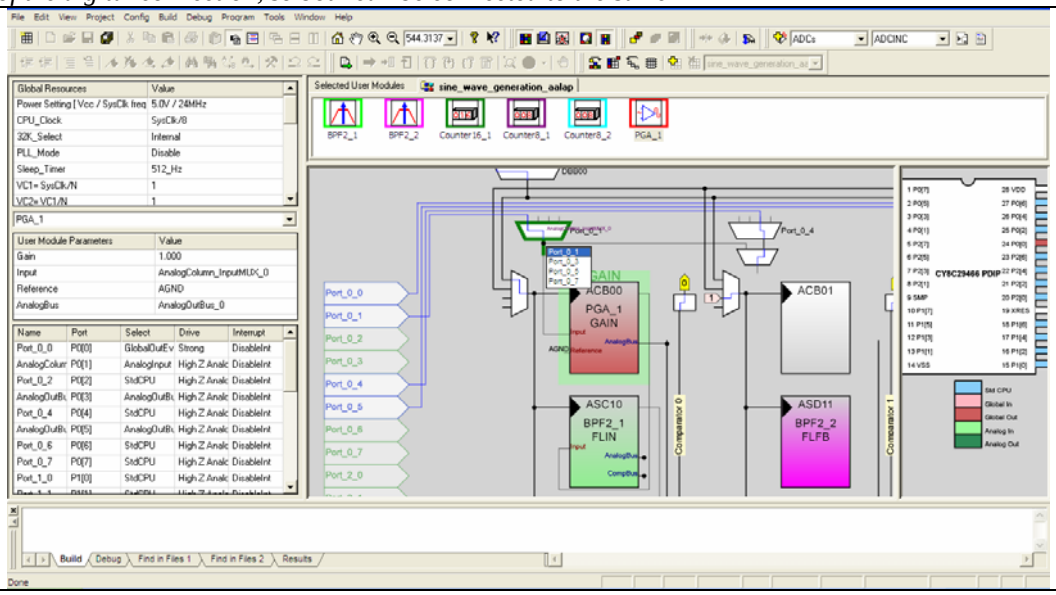


Connect the CompareOut to RO0[0]. Then to Global Out Even (GOE) 0 and then to Port_0_0. Other ports might be used. I

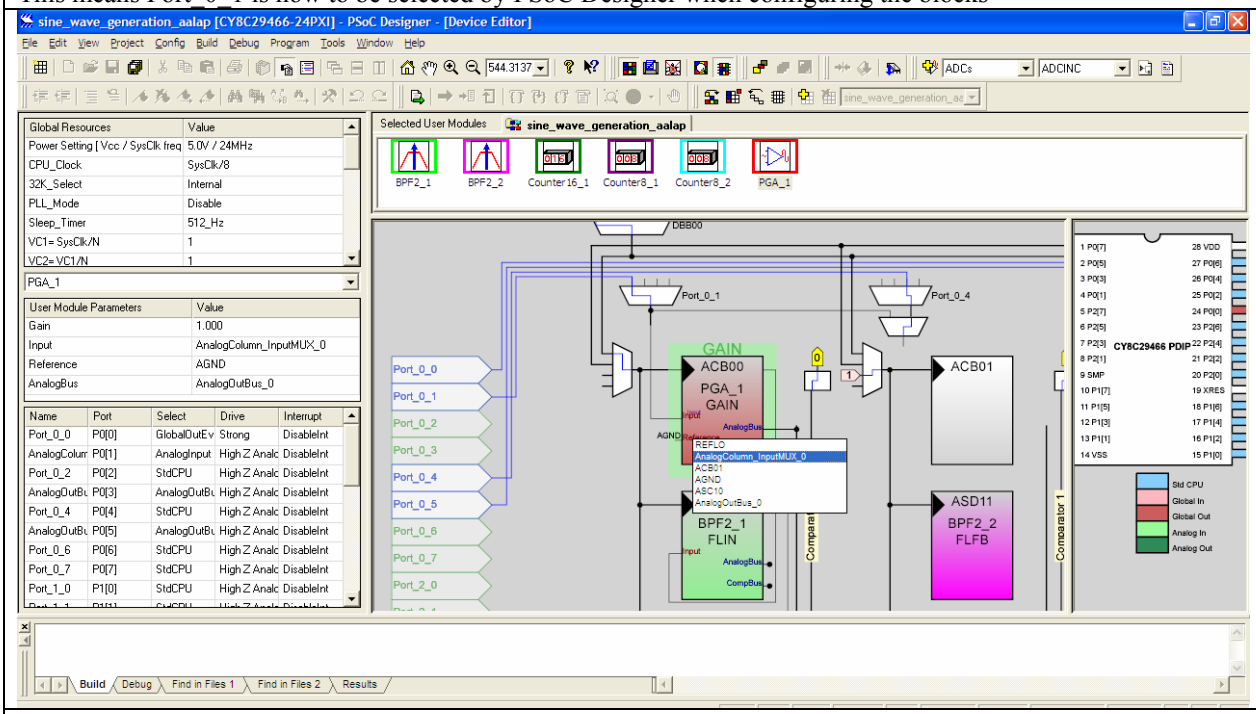
have used this for convenience of the external connections which I propose to use.

Remember to connect Port_0_0 to Port_0_1 because this is what I am assuming from the next step onwards. That is the square wave generated will be available at Port_0_1 now.

Note: You can connect the digital output to Port0.0 and route that into the analog input. The analog connection is independent of the digital connection, so both can be connected to the same

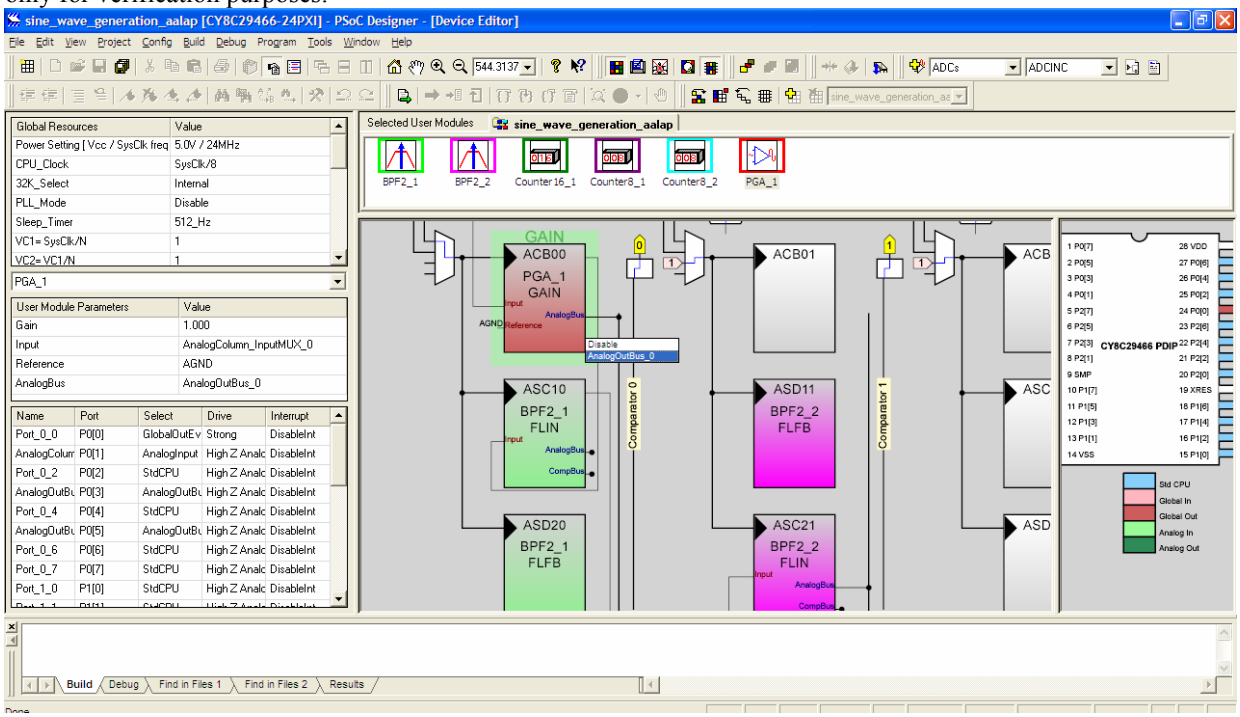


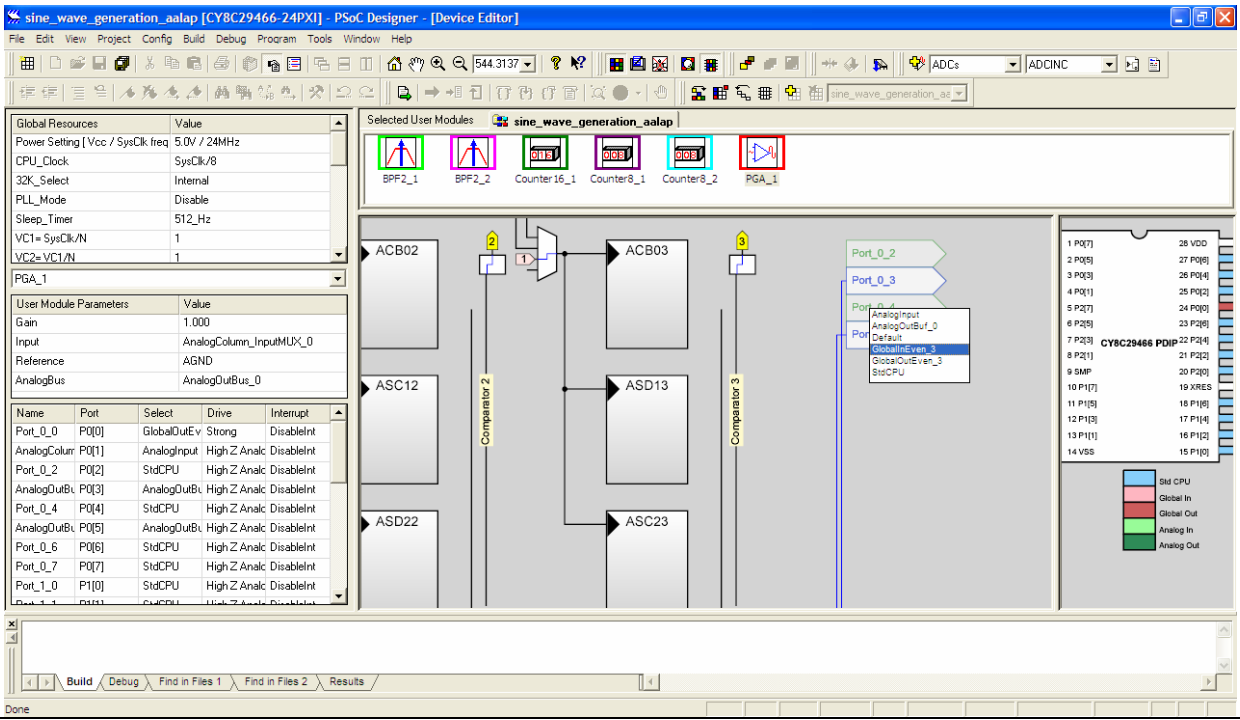
13. Scroll down to the AnalogColumns_InputMux0 and select Port_0_1.
This means Port_0_1 is now to be selected by PSoC Designer when configuring the blocks



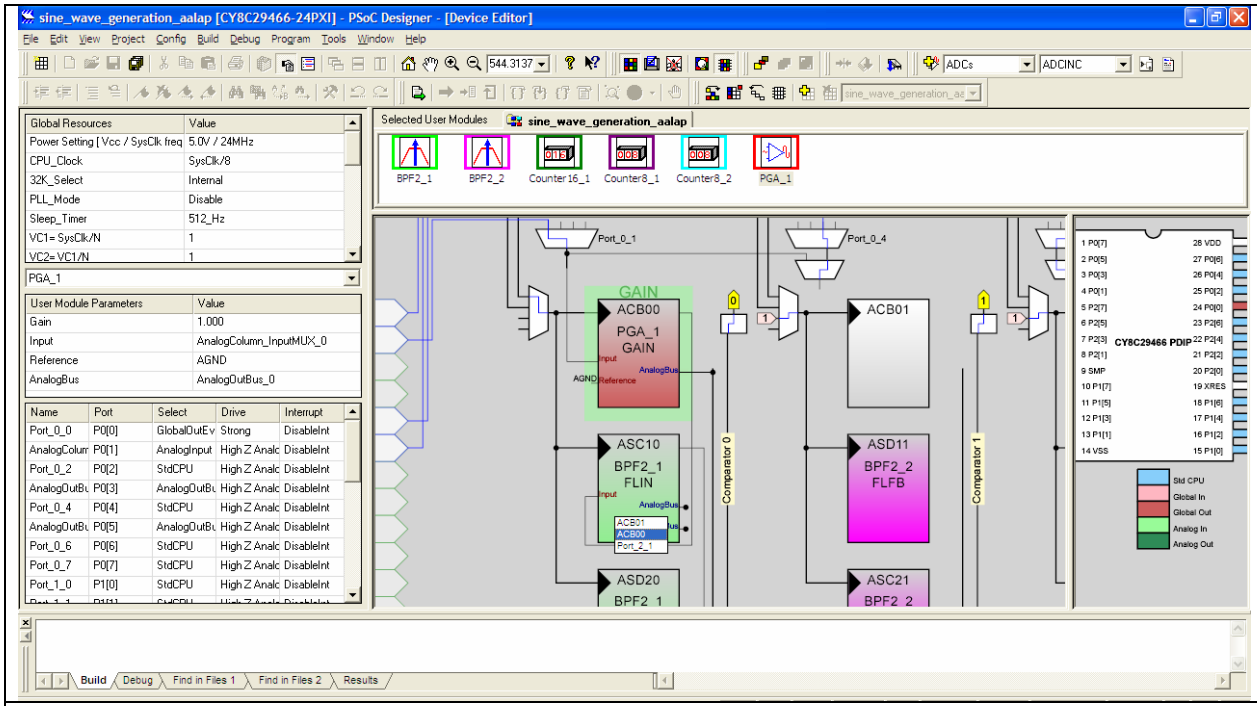
14. Now select AnalogColumns_InputMux0 as the Input to the PGA block.
This effectively makes the input at Port_0_1 of the PSoC available as input

One can also connect the AnalogBus to AnalogOutBus_0. This can be then connected to a Pin as shown below. This step is only for verification purposes.





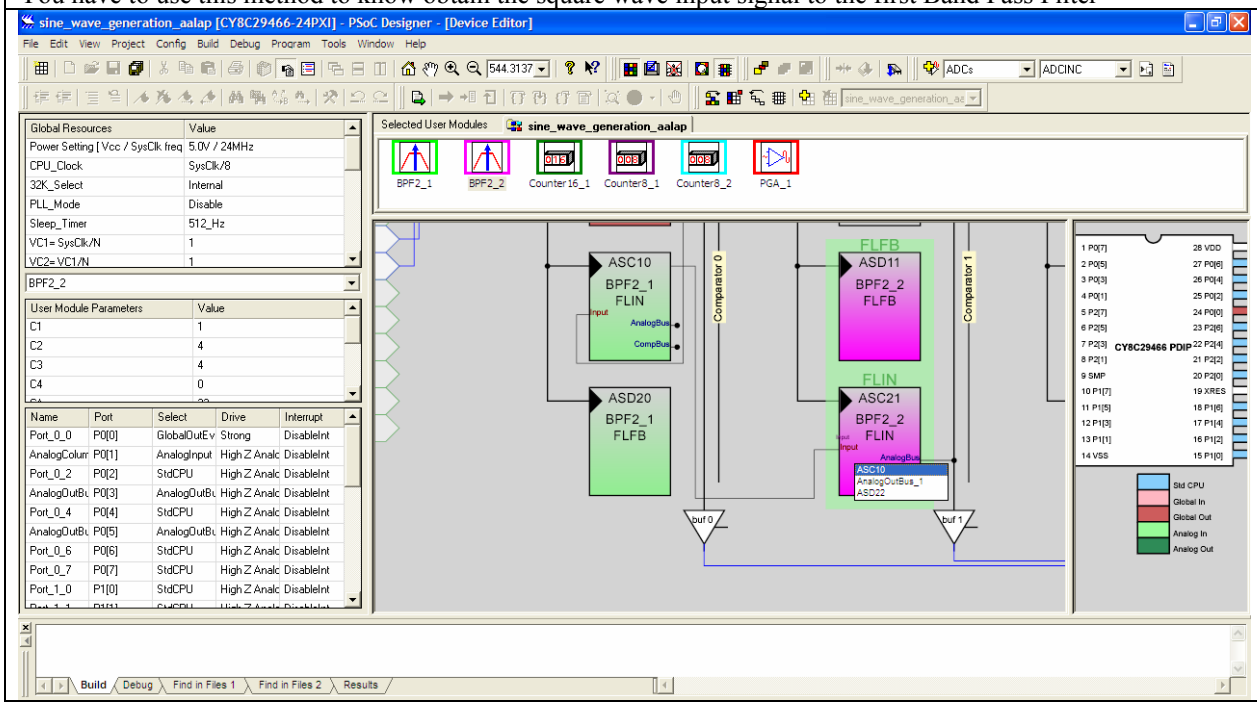
15. Once we get a suitable analog value from the PGA, we need to feed it to the BPF.
For the first iteration, one may use the part placement as shown in the figure shown next – other alternatives have their own problems.



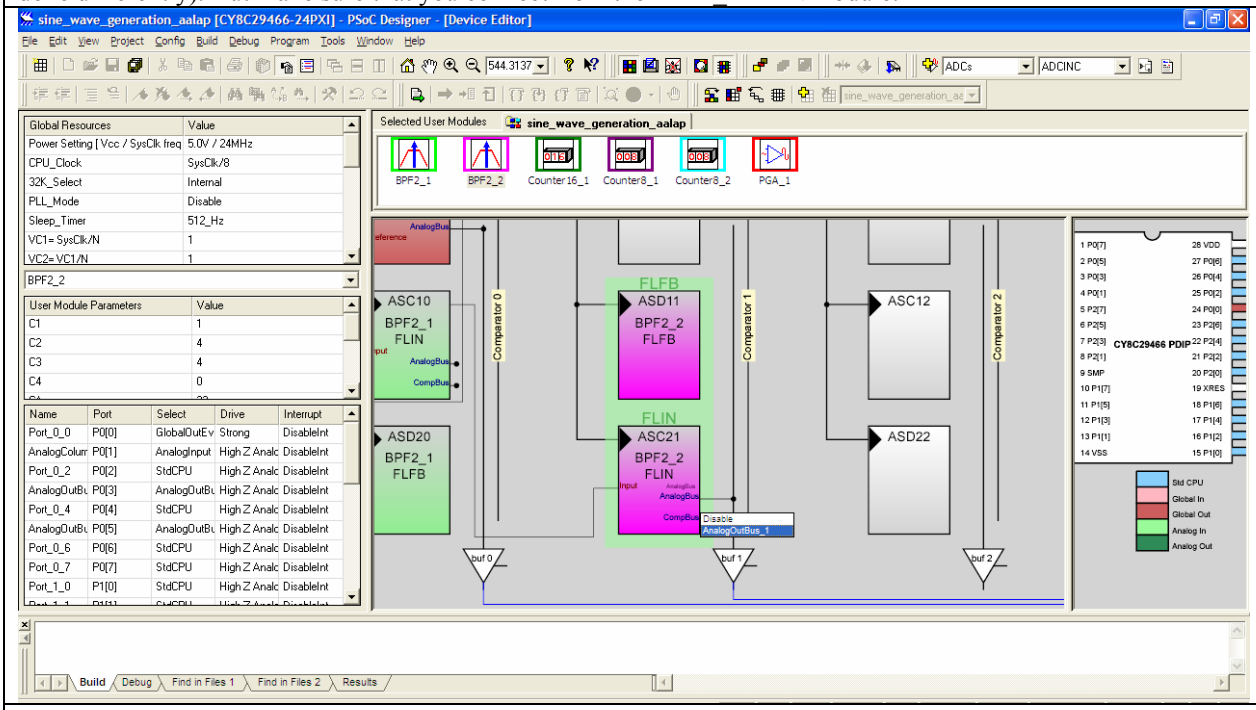
16. Click on the input of BPF2_1 FLIN Module and select ACB00 (it might be something different if you placed it differently)
Te opposite connection that is from PGA to BPF2_1 is not generally used (or possible!!)

Note : The way to set analog connections is by selecting which source is used for each input. Where the output goes cannot be set. This is just the way that PSoC Designer was made to work.

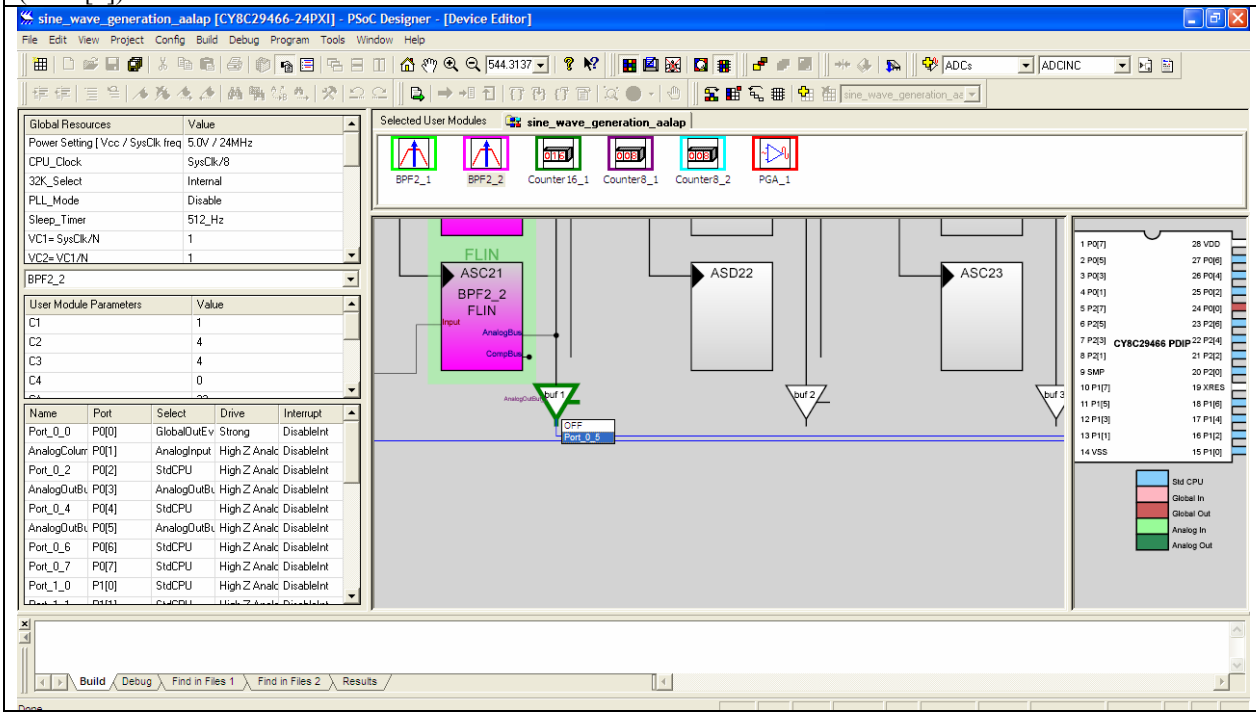
You have to use this method to know obtain the square wave input signal to the first Band Pass Filter



17. Click on the Input of BPF2_2 module and select ASC10 (again this name might be different if your placement has been done differently). But make sure that you connect from the BPF2_1 FLIN module.



18. To obtain the final output (now a sine wave) connect the AnalogBus of the BPF2_2 to the AnalogOutBus_1
19. You will notice that this is fed to buf1. Click on this to connect to Port0_5. Now the output can be sampled from Pin2 (Port0[5]) of the PSoC

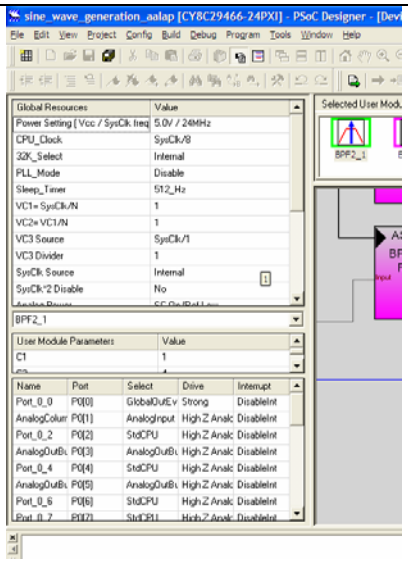


20. Global Resources Settings
    - 32K_Select and PLL_Mode at Internal and Disable
    - We are not using an external crystal to drive the processor nor need anything to sync it to.
    - CPU_Clock=SysClock/8 (=3MHz)
    - VC1=1 (We don't need this)
    - VC2=1
    - VC3 Source = SysClk/1 (Default)
    - VC3 Divider=1
    - Every other parameter default

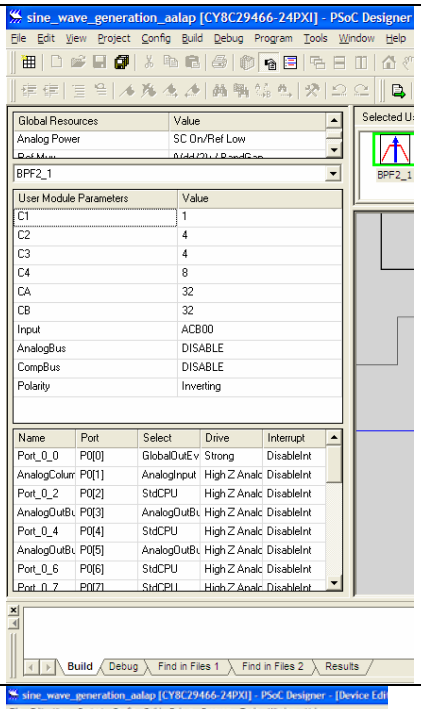
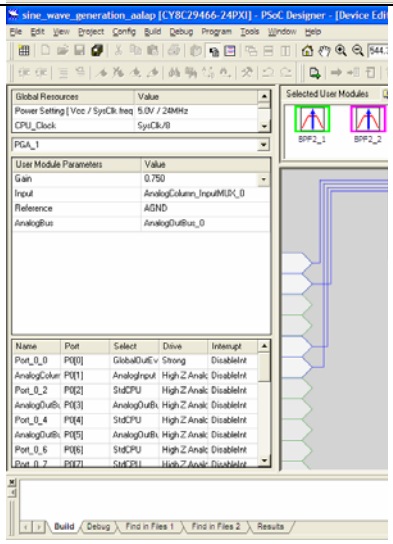
- *Notes : The "best" speed to set the CPU for experimental use is 12MHz. That is the maximum speed over the full voltage range (using 24MHz requires >4.75V). If projects require minimum power, they can have their CPU speed reduced after it gets working.*

- *Jeff recommends using VC1, VC2 or VC3 as the source for the clock for the square wave instead of the 32k Clock.*

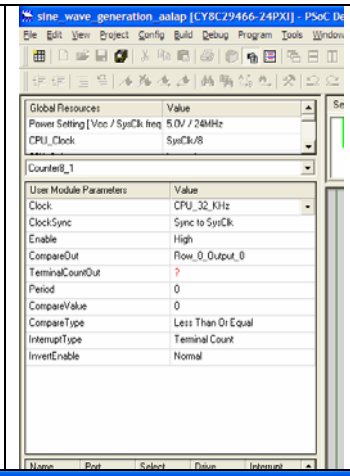
- *Generally, when clocks are not being used, they should be set to the lowest frequency (e.g. VC1 = 16, VC2 = 16, VC3 Source = VC2, VC3 = 256). This will use the least power.*

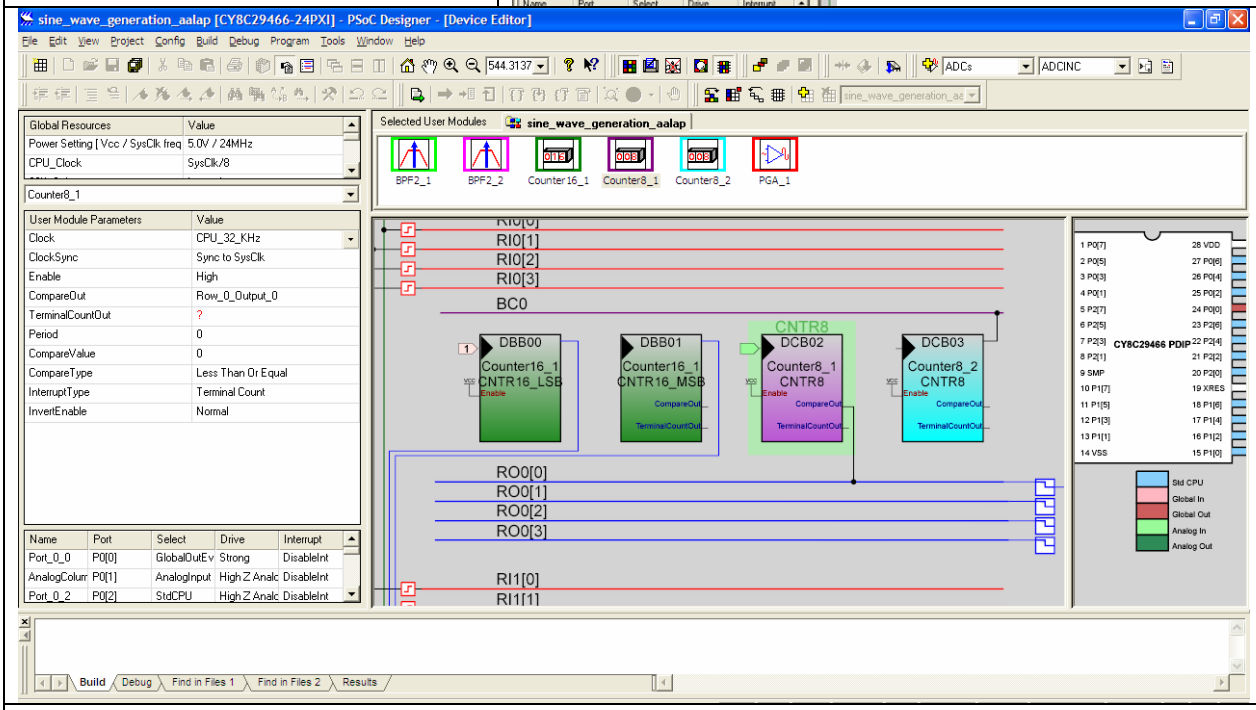


(All Default)

21. Refer to the Band Pass Filter Design Utility. The following parameters can be used. *Note : The Wizard does not currently work properly. Cypress plans on having it fixed in a future release.*	
22. The following parameters for the Programmable Gain Amplifier should be used. • For secondary testing, the gain can be varied here and experimented *Notes :* • *This is prior to the filter, so only setting gains <1 should have an effect. Another option would be to change the gain in the Filter.* • *You can have PGA <1 (like suggested above) and then change the gain of the BPF itself to see the effect on amplitude.* • *The gain on the BPF is negative (not obvious with a sine wave output).*	

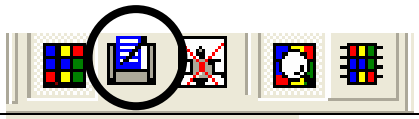
23. The User Module Parameters shown alongside for the 8 bit counter should be used. One can experiment with different values once the primary result has been obtained.





Counter8_2 is to show how broadcast buses (BC0) here can be used to take the output of one module can be fed to another. This was actually used in AN 2086 to provide control of the frequency of the sine wave generated using a digital encoder. For more information, refer the appendix.

21. **Shift to Application Editor**



22. In application editor, Press **F7** or **Build|Build All** from the menu
    - This step generates all the files associated with the user module we have selected and update header files and libraries as well.
    - In the left top pane, observe all the files created and make a note.

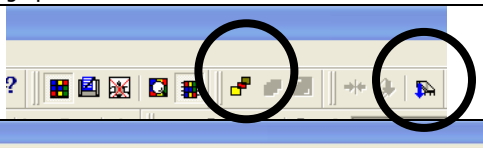
- With more components added, more files will be created.

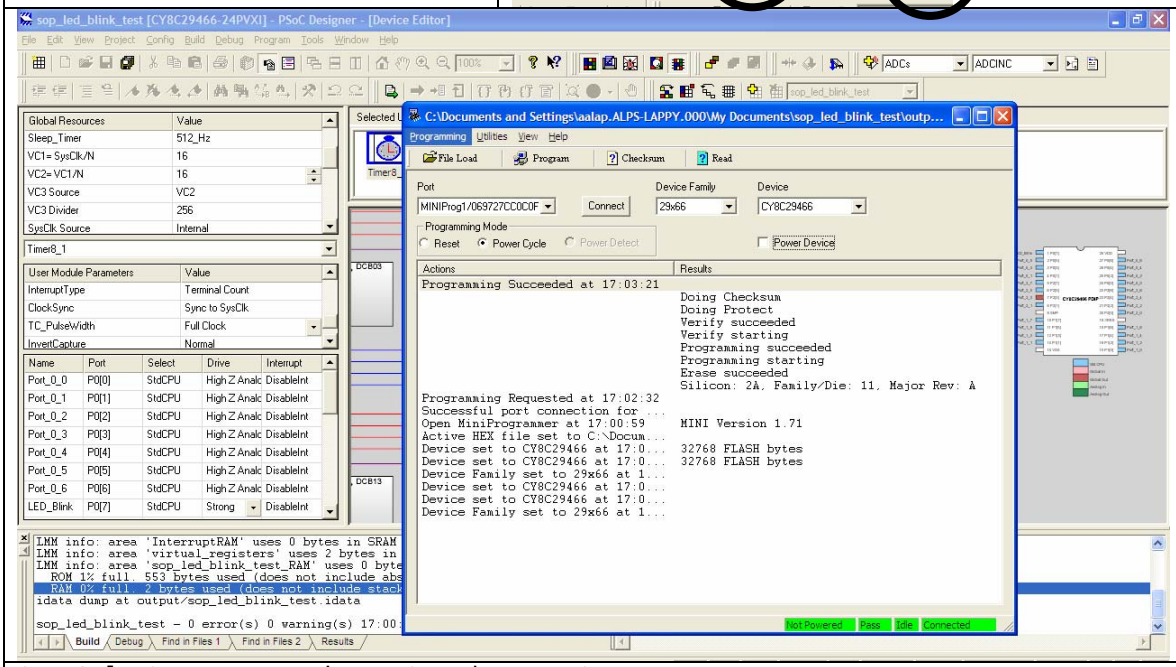
21 Type the following code in main.asm

```
; Assembly main line

include "m8c.inc"       ; part specific constants and
macros
include "memory.inc"    ; Constants & macros for SMM/LMM
and Compiler
include "PSoCAPI.inc"   ; PSoC API definitions for all
User Modules
export _main
export flags, ticker, period
area   bss    (ram)                    ;inform assembler
of variables to follow
area text (ROM, REL)
_main:
       M8C_EnableGInt
       call   Counter16_1_Start
       call   Counter8_1_Start
       call   Counter8_2_Start  ;Turn on Ticker
       call   Counter8_2_EnableInt
       mov          a,3;bPowerSetting to HighPower Mode.
Refer Datasheet
       call   BPF2_1_Start
       mov          a,3;bPowerSetting to HighPower Mode.
Refer Datasheet
       call   BPF2_2_Start
       mov          a,3
       call   PGA_1_Start    ;Turn on buffer
.terminate:
jmp .terminate
```

22. Build
23 Press Program Part on the right



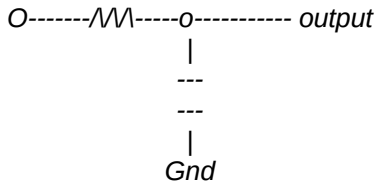


24. Select MINIProg1 in Port and connect

```
25. Select Program
26. Make sure you check “power device” icon after “Programming Successed” is displayed
27. Make sure Port0[5] is connected to a CRO!!
```

Modifications/Exericse :

1. Connect the output of BPF2_1 FLIN module to the analog bus, then via the buffer to a pin of your choice. Observe the difference if any between the outputs of the two Band Pass Filters. This will enable us to understand why at all two BPF Filters should be used.
2. Try giving the output of the 8 bit counter directly to the Band Pass Filter (Internally and externally both). You will notice that the input may also be given through a buffer amplifier. Try changing the gain of the buffer or PGA (to 1, then to higher values) and observe the changes in the sine wave output waveforms.
3. Perform the application mentioned in AN2086. In case a digital encoder is not available, use a microprocessor or another PSoC to generate the output waveforms mentioned in the Application Note.
4. *An important thing to do is to add an R-C LPF on the output.*



The purpose of this filter is to remove the sample clock from the SC block. Its pole should be set between the pass frequency of the BPF and the frequency of the SC blocks. The pole for the RC must be higher than the BPF frequency but has to be low enough so that the SC clock is removed sufficiently. Look in the spreadsheet to see what the oversample frequency of the BPF is. This will give an idea of the limits for the RC LPF.

Review of this Experiment :

1. Note : Items in italics refer to suggestions on this experiment by Jeff Dahlin, Principal Applications Engineer, Cypress Semiconductors, San Jose. He may be reached for concrete doubts on jvy@cypress.com. Please first use the Developer Forums at psocdeveloper.com for queries before contacting Jeff.
2. This experiment has been reviewed by Jeff Dahlin, PAE, Cypress Semiconductors.